

И. Н. ОГОРОДНИКОВ

ВВЕДЕНИЕ В ОБРАТНЫЕ ЗАДАЧИ ФИЗИЧЕСКОЙ ДИАГНОСТИКИ. МОДЕЛЬНЫЕ РАСЧЕТЫ В МАТЛАБ

Учебное пособие

Министерство образования и науки Российской Федерации

Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина

И. Н. Огородников

**Введение в обратные задачи
физической диагностики.
Модельные расчеты в Матлаб**

Учебное пособие

*Рекомендовано методическим советом
Уральского федерального университета
для студентов вуза, обучающихся
по направлениям подготовки магистров
14.04.02 «Ядерная физика и технологии»,
14.04.04 «Биотехнические системы и технологии»
и специальности 14.05.04 «Электроника и автоматика
физических установок»*

Екатеринбург
Издательство Уральского университета
2017

УДК 517.9:001.891.57(076.5)

ББК 22.1я73-5+22.3я73

О-39

Рецензенты:

лаборатория математического моделирования в экологии и медицине Института промышленной экологии Уральского отделения Российской академии наук (канд. техн. наук, доц. *В. Г. Панов* и д-р физ.-мат. наук, проф. *А. Н. Вараксин*);

д-р физ.-мат. наук, гл. науч. сотр. Института физики металлов Уральского отделения Российской академии наук *В. И. Соколов*

Научный редактор — д-р физ.-мат. наук, проф. *А. Н. Кислов*

Огородников, И. Н.

О-39 Введение в обратные задачи физической диагностики. Модельные расчеты в Матлаб : учеб. пособие / И. Н. Огородников. — Екатеринбург : Изд-во Урал. ун-та, 2017. — 128 с. ISBN 978-5-7996-2261-9

Учебное пособие направлено на развитие у студентов практических навыков создания теоретических и математических моделей и методов расчета современных физических установок, приборов радиационной безопасности и окружающей среды, а также различных приборов биофизического и медицинского назначения.

Пособие предназначено для студентов технических специальностей физико-технологического института всех уровней обучения и соответствует федеральному государственному образовательному стандарту третьего поколения.

Библиогр.: 11 назв. Табл. 10. Рис. 29.

УДК 517.9:001.891.57(076.5)

ББК 22.1я73-5+22.3я73

ISBN 978-5-7996-2261-9

© Уральский федеральный университет, 2017

Оглавление

Предисловие	5
1. Оптимизация и теория двойственности	7
1.1. Двойственность в оптимизации	7
1.2. Основы табличного симплексного метода	12
1.3. Решение прямой и двойственной задач	19
1.3.1. Постановка задачи	19
1.3.2. Формулировки прямой и двойственной задач	19
1.3.3. Симплекс-метод решения прямой задачи	21
1.3.4. Оптимальное решение двойственной задачи	26
1.4. Решение задач оптимизации в среде Матлаб	27
1.4.1. Функция linprog	27
1.5. О недостатках симплексного метода	30
1.6. Выполнение численного расчета	31
1.7. Упражнения для самоконтроля	32
2. Регуляризация на компактных множествах	33
2.1. Понятие компактных пространств	33
2.1.1. Некорректные задачи на компактах	34
2.2. Свойства метода регуляризации на компактных множествах	37
2.3. Конечно-разностные аналоги компактных множеств в L_2	42
2.4. Основы построения алгоритма регуляризации	47
2.5. Формулировка модельной задачи	52
2.6. Описание программного обеспечения	56
2.6.1. Минимизация методом условного градиента	56
2.6.2. Выбор оптимальной вершины многогранника	58
2.6.3. Интерфейс к программе Compact	59
2.6.4. Вывод результатов тестового расчета	62
2.7. Выполнение численного расчета в среде Матлаб	63
3. Регуляризация по методу обобщенной невязки	67
3.1. Метод регуляризации А.Н. Тихонова	67
3.2. Регуляризация для уравнения Фредгольма	72
3.3. Регуляризация численного дифференцирования	78
3.4. Программное обеспечение расчетов в среде Матлаб	79

3.5. Выполнение численного расчета в среде Матлаб	88
4. Итеративная регуляризация вариационных неравенств . .	91
4.1. Сведения из теории вариационного исчисления	91
4.2. Элементы теории вариационных неравенств	96
4.2.1. Принцип итеративной регуляризации	100
4.3. Пример решения простейшей задачи	102
4.4. Выполнение практического расчета в среде Матлаб . . .	110
Список библиографических ссылок	112
Алфавитный указатель	113
Приложение 1. Упражнения для расчетов в среде Матлаб . .	116
Приложение 2. Вычисление интегралов в среде Матлаб . . .	123

Предисловие

Учебное пособие основано на семестровом курсе лекций с условным названием «Специальные главы высшей математики (Введение в обратные задачи физической диагностики)», который читается в Уральском федеральном университете для студентов старших курсов физико-технологического института, специализирующихся в электронике и автоматике физических установок, защите от излучений, радиационной экологии, биомедицинской инженерии, а также в конструировании приборов для применения в области радиационной безопасности человека и окружающей среды. Оно посвящено реализации расчетных методов, изложенных в [1], и может быть полезно для студентов других технических специальностей.

Цель учебного пособия — помочь студентам в овладении практическими навыками применения некоторых инструментов решения обратных задач физической диагностики. Такая формулировка цели предопределила направленность выбора материала для данного пособия, поэтому при его составлении во главу угла было положено несколько основополагающих принципов.

Во-первых, в данном пособии не обсуждаются результаты полного и всестороннего математического исследования обсуждаемых методов решения обратных задач. В рамках сформулированного принципа все необходимые теоремы в пособии приведены лишь в качестве справочного материала: их число минимизировано и они приводятся без доказательства. Однако для каждой теоремы приведена точная ссылка на первоисточник (с указанием страниц), где содержится полное и всестороннее математическое исследование обсуждаемого вопроса. Вдумчивый и подготовленный читатель без труда найдет эту информацию в литературе, цитируемой в приведенном библиографическом списке.

Во-вторых, для демонстрации обсуждаемых методов решения обратных задач и разработки инструментов их практического решения была выбрана математическая среда универсального назначения, а не какое-либо специализированное программное обеспечение. Причины такого выбора определяются учебной направленностью пособия: специализированное программное обеспечение имеет, как правило, ограниченную область применения и, самое главное, является «непрозрачным» с точки зрения алгоритмов и методов, применяемых в нем для решения поставленных задач. Напротив, универсальная математиче-

ская среда широкого применения позволяет продемонстрировать все особенности алгоритмов и методов решения обратных задач.

В настоящее время для профессионального и учебного применения доступен широкий набор математических сред универсального назначения. Не ставя задачу сравнительного анализа этих сред, отметим лишь, что из широкого набора доступных математических сред универсального назначения была выбрана *Matlab*[®] — матричная лаборатория фирмы *The Math Works*[™] (далее для краткости — Матлаб).

В-третьих, данное учебное пособие не направлено на детальное исследование и обсуждение всех аспектов программирования обсуждаемых методов решения обратных задач. В связи с этим, при составлении учебных программ использован лишь минимальный набор средств программирования. Для понимания особенностей функционирования большинства приведенных программ (или фрагментов программного кода) требуется владение лишь базовым уровнем языка программирования Матлаб. Однако учебное пособие не ставит перед собой задачу обучения основам работы (и программирования) в среде Матлаб, оставляя этот аспект для специализированных учебных курсов и многочисленной литературы.

На страницах учебного пособия приведены и обсуждаются исходные тексты головных программ, применяемых для демонстрации соответствующих методов решения. Для их функционирования используются также многочисленные вспомогательные модули, реализующие рутинные процедуры. Весь программный код на языке Матлаб создан автором целенаправленно для использования в учебном пособии, однако полный объем программного кода, подготовленного для достижения поставленной цели, оказался все же достаточно обширным, чтобы его можно было привести в виде исходных текстов на страницах учебного пособия. Программное обеспечение для демонстрации возможностей обсуждаемых методов может быть бесплатно выслано по электронной почте по запросу. Запрос можно сделать по адресу inv_illposed@bk.ru. Общий объем программного обеспечения, сжатого в zip-архив, не превышает 1 МБ. Разархивированные программы не требуют инсталляции.

Программные модули расположены в архиве по главам. Для каждой главы в архиве имеется файл Contents, в котором перечислены программы и используемые ими служебные модули. Для головных модулей, которые нужно запускать в Матлаб, принято соглашение: их названия начинаются с прописной буквы. Напротив, названия служебных модулей начинаются со строчных букв. Нет никаких лицензионных ограничений на использование и модификацию приведенных в архиве модулей для решения собственных задач читателя.

1. Оптимизация и теория двойственности

1.1. Двойственность в оптимизации

Понятие *двойственности* в математическом программировании носит фундаментальный характер, оно имеет важное теоретическое и практическое значение. В рамках этого понятия формулируют две сопряженные задачи: *прямая задача* (П) — исходная задача линейного программирования и *двойственная задача* (Д) — сопряженная задача, которую получают из прямой задачи с помощью установленного набора правил [2].

1. Основополагающая идея теории двойственности. Для любой исходной задачи (П) можно сформулировать сопряженную задачу (Д). Между решениями этих задач имеется тесная связь. Количественное выражение связи представлено рядом соотношений, которые используют при анализе моделей на чувствительность, исследовании свойств оптимального решения и проверке оптимальности допустимого решения.

Определение. Две экстремальные задачи на нахождение оптимальных решений считаются *эквивалентными* задачами, если множества их решений совпадают, или обе задачи не имеют решений.

Формулировка прямой задачи в значительной мере определяется типами ограничений, знаками переменных (неотрицательные или свободные, т.е. без ограничения в знаке) и типом оптимизации (максимизация или минимизация целевой функции). Теорию двойственности часто излагают, рассматривая формулировки двойственных задач в зависимости от особенностей формулировки прямых задач. Для унификации правил перехода к двойственной задаче выработали стандартную формулировку исходной прямой задачи. В стандартной форме задача линейного программирования записывается следующим образом: максимизировать или минимизировать целевую функцию z при заданных ограничениях b_i , $i = \overline{1, m}$:

$$z = \sum_{j=1}^n c_j x_j; \quad \sum_{j=1}^n a_{ij} x_j = b_i, \quad x_j \geq 0, \quad j = \overline{1, n}.$$

Список n переменных x_j может включать также некоторые дополнительные (фиктивные) переменные. Для записи задачи линейного программирования в стандартной форме необходимо выполнить следующие обязательные условия.

1. Все ограничения задачи записаны в виде ограничений-равенств с неотрицательной правой частью. Отметим, что если среди ограничений задачи имеются ограничения-неравенства, то путем регулярных преобразований они могут быть сведены к ограничениям-равенствам.

2. Все переменные задачи считаются неотрицательными.

3. Оптимизация — это задача на экстремум целевой функции, она определяется как минимизация или максимизация целевой функции.

Для стандартной формулировки исходной задачи линейного программирования используют стандартизованную таблицу симплекс-метода. На основе стандартной формы прямой задачи формулируют двойственную задачу. При этом ограничения и переменные двойственной задачи получают используя унифицированные правила симметричных структурных преобразований прямой задачи. После вычислений симплекс-метода получаем решение прямой задачи. Непосредственно из симплекс-таблицы, соответствующей оптимальному решению прямой задачи, можно автоматически получить решение двойственной задачи [3, 4].

2. Правила перехода к двойственной задаче. Существуют следующие правила для перехода от прямой задачи к двойственной:

а) ограничения-неравенства в исходной задаче (П) записывают в стандартном виде со знаками: $[\geq]$ (min-процедура) и $[\leq]$ (max-процедура);

б) каждому из m ограничений прямой задачи (П) ставится в соответствие переменная двойственной задачи (Д);

в) каждой из n переменных прямой задачи (П) ставится в соответствие ограничение двойственной задачи (Д);

г) пусть x_j — переменная прямой задачи (П), тогда коэффициенты при x_j в ограничениях задачи (П) становятся коэффициентами ограничения задачи (Д), соответствующей этой переменной, а правая часть формируемого ограничения будет равна коэффициенту при x_j в целевой функции;

д) правые части ограничений прямой задачи (П) становятся коэффициентами целевой функции двойственной задачи (Д);

е) если переменная задачи (П) является неотрицательной ($x_j \geq 0$), то j -е условие системы ограничений задачи (Д) является неравенством в соответствие п. а;

ж) ограничение j двойственной задачи записывается в виде строгого равенства, если на переменную x_j прямой задачи не наложено условие неотрицательности;

з) если в прямой задаче (П) имеются ограничения равенства, то на соответствующие переменные двойственной задачи (Д) не накладывается условие неотрицательности.

В графическом виде эти правила представлены в табл. 1.1.

Таблица 1.1

Соответствие прямой и двойственной задач

Переменные двойственной задачи	Переменные прямой задачи						Коэффициенты целевой функции двойственной задачи
	x_1 c_1	x_2 c_2	\dots \dots	x_j c_j	\dots \dots	x_n c_n	
y_1	a_{11}	a_{12}	\dots	a_{1j}	\dots	a_{1n}	b_1
y_2	a_{21}	a_{22}	\dots	a_{2j}	\dots	a_{2n}	b_2
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
y_m	a_{m1}	a_{m2}	\dots	a_{mj}	\dots	a_{mn}	b_m

j -ограничение
задачи Д

Далее удобно перейти к матричной форме записи. Для этого введем новые обозначения:

$C = [c_j]$ – вектор-строка $(1 \times n)$ коэффициентов c_j ;

$X = [x_j]$ – вектор-столбец $(n \times 1)$ переменных x_j ;

$A = [a_{ij}]$ – матрица $(m \times n)$ ограничений;

$B = [b_i]$ – вектор-столбец $(m \times 1)$ свободных членов (правая часть) для матрицы ограничений;

$U = [u_i]$ – вектор-строка $(1 \times m)$ промежуточных переменных u_i .

В таком случае прямую задачу можно записать в матричной форме

$$\max z = C \times X, \quad A \times X \leq B, \quad X \geq 0. \quad (1.1)$$

Формулировка двойственной задачи (Д) вытекает из формулировки прямой задачи (П): для перехода к двойственной задаче (Д) следует применить табл. 1.2 и учесть, что вектор-столбец Y $(m \times 1)$ переменных задачи (Д) связан с промежуточной переменной U соотношением $Y = U^T$, табл. 1.2.

Для прямой задачи (1.1) формулировка двойственной задачи (Д) заключается в минимизации целевой функции w при ограничениях A :

$$\min w = U \times B, \quad U \times A \geq C, \quad U \geq 0. \quad (1.2)$$

Таблица 1.2

Правила перехода к двойственной задаче

Прямая задача	Двойственная задача
Целевая функция (\min)	Правая часть ограничений
Правая часть ограничений	Целевая функция (\max)
A-матрица ограничений	A^T -матрица ограничений
Ограничение $b_i \geq 0, (\leq 0)$	Переменная $u_i \geq 0, (\leq 0)$
Ограничение $b_i = 0$	Переменная $u_i \neq 0$
Переменная $x_j \geq 0$	j -ограничение: " \leq "
Переменная $x > 0$	j -ограничение: " $=$ "

Чтобы получить окончательную формулировку двойственной задачи, необходимо перейти от промежуточных переменных U к переменным Y двойственной задачи:

$$\min w = B^T \times Y, \quad A^T \times Y \geq C^T, \quad Y \geq 0. \quad (1.3)$$

Отметим, что если задачу (1.3) принять за прямую задачу, то формулировка задачи, двойственной к ней, совпадет с исходной задачей (1.1).

Рассмотрим пример формулировки двойственной задачи из прямой.

Задача (П): $\min z = 2x_1 - 3x_2$, ограничения заданы системой:

$$\left\{ \begin{array}{l} x_1 - x_2 \leq 1; \\ 2x_1 + 3x_2 \geq 4; \\ x_1 + x_2 = 3; \\ x_1 \geq 0; \quad x_2 > 0. \end{array} \right. \Rightarrow \left\{ \begin{array}{l} -x_1 + x_2 \geq -1, \quad b_1 = -1; \\ 2x_1 + 3x_2 \geq 4, \quad b_2 = 4; \\ x_1 + x_2 = 3, \quad b_3 = 3; \\ x_1 \geq 0; \quad x_2 > 0. \end{array} \right.$$

Очевидно, что переменная x_2 не ограничена по знаку, т. е. она может быть как меньше, так и больше нуля. Сформулируем двойственную задачу: поскольку число ограничений равно трем, то получим три переменные u_1, u_2, u_3 . Целевая функция двойственной задачи (Д) будет иметь вид:

$$\max w = U \times B = u_1 b_1 + u_2 b_2 + u_3 b_3 = -u_1 + 4u_2 + 3u_3.$$

Составим матрицы для коэффициентов ограничений обеих задач:

$$\begin{array}{ll} \text{Прямая задача (П)} & \text{Двойственная задача (Д)} \\ A = \begin{bmatrix} -1 & 1 \\ 2 & 3 \\ 1 & 1 \end{bmatrix}; & A^T = \begin{bmatrix} -1 & 2 & 1 \\ 1 & 3 & 1 \end{bmatrix}. \end{array}$$

Задача (Д). Формулировка двойственной задачи при этом будет иметь следующий вид:

$$\begin{aligned} \max w &= -u_1 + 4u_2 + 3u_3. \\ \begin{cases} -u_1 + 2u_2 + u_3 \leq 2, & c_1 = 2; \\ u_1 + 3u_2 + u_3 = -3, & c_2 = -3. \\ u_1 \leq 0; & u_2 \geq 0; & u_3 > 0. \end{cases} \end{aligned}$$

Задачи (П) и (Д) являются равноправными: любую из них можно выбрать в качестве прямой задачи (П), тогда другая задача будет считаться двойственной. С понятием двойственности связаны три важные теоремы.

Теорема 1 (двойственности) [5, с. 132–133]. *Если X и U – соответственно допустимые решения прямой (1.1) и двойственной (1.3) задач, то*

$$Z = C \times X \leq W = U \times B.$$

Из теоремы 1 следует, в частности, что значение целевой функции прямой задачи (П) не превосходит значения целевой функции двойственной задачи (Д).

Если решения прямой и двойственной задач \bar{X} , \bar{U} удовлетворяют равенству $C \times \bar{X} = \bar{U} \times B$, то планы \bar{X} и \bar{U} являются оптимальными решениями прямой (П) и двойственной (Д) задач соответственно.

Теорема 2 [5, с. 133–135]. *Пусть заданы прямая (П) и двойственная (Д) задачи:*

а) если прямая (1.1) и двойственная (1.3) задачи имеют решения, то каждая из этих задач имеет оптимальное решение и

$$\bar{Z} = \min(\Pi) = \max(\Delta) = \bar{W}$$

б) если одна из них имеет неограниченный оптимум, то другая не имеет решения.

Теорема 3 [5, с. 136–137]. Если вектор-столбцы $\bar{X} = (\bar{x}_1, \dots, \bar{x}_n)$ и $\bar{Y} = (\bar{y}_1, \dots, \bar{y}_m)$ являются оптимальными решениями прямой (1.1) и двойственной (1.3) задач линейного программирования, то:

$$\begin{aligned} \left(\sum_{j=1}^n a_{ij} \bar{x}_j - b_i \right) \bar{y}_i &= 0, \quad i = \overline{1, m}; \\ \left(\sum_{i=1}^m a_{ij} \bar{y}_i - c_j \right) \bar{x}_j &= 0, \quad j = \overline{1, n}. \end{aligned}$$

1.2. Основы табличного симплексного метода

Одним из основополагающих методов исследования задач линейного программирования является симплекс-метод. Он базируется на т. н. фундаментальной теореме симплекс-метода, которая утверждает, что если задача линейного программирования записана в канонической форме, тогда среди её оптимальных планов обязательно будет опорное решение системы ограничений этой задачи. Если задача имеет единственный оптимальный план, то он обязательно будет совпадать с одним из опорных решений системы ограничений этой задачи.

На основании фундаментальной теоремы симплекс-метода исследование задачи линейного программирования может быть сведено к исследованию системы линейных ограничений этой задачи, которая в пространстве переменных задает многогранное множество. Линейность целевой функции $Z(X) = c_1 x_1 + c_2 x_2 + \dots c_n x_n + \dots c_{n+m} x_{n+m}$ подразумевает, что она не имеет экстремума внутри этого многогранного множества, т. е. экстремум $Z(\bar{X})$ достигается только на границе области допустимых решений в одной из угловых точек многогранного множества. Из ограниченности линейной целевой функции на многограннике допустимых решений задачи линейного программирования вытекает, что существует такая угловая точка многогранника решений, в которой линейная целевая функция задачи достигает своего оптимума, а каждый опорный план соответствует угловой точке многогранника решений этой задачи.

Для решения задачи линейного программирования нужно исследовать только вершины многогранника решений, т. е. только опорные планы.

Рассмотрим для начала любую из вершин многогранника условий системы ограничений и применим к ней критерий оптимальности. Если выбранная вершина не удовлетворяет критерию минимума (максимума), то необходимо перейти к другой вершине так, чтобы значение

в задаче на минимум или максимально в задаче на максимум. Однако для этого нужно найти все опорные решения, которые заранее неизвестны. В реальных задачах количество этих решений может быть велико, что затрудняет их полный перебор. В симплекс-методе реализована процедура *направленного* перебора опорных решений. В качестве стартовой точки здесь используют любое заранее найденное опорное решение. Далее применяют алгоритм симплекс-метода и находят новое опорное решение, для которого значение целевой функции $Z(\bar{X})$ будет не больше (минимум) или не меньше (максимум), чем на старом решении. Для достижения оптимального плана может потребоваться несколько таких шагов (итераций).

Симплекс-метод устанавливает формальный порядок в нахождении первого (исходного) базисного решения и алгоритм перехода к другим базисным решениям. Для этого систему ограничений приводят к общему виду, то есть к системе m линейных уравнений с $n + m$ переменными ($m < n$). Затем находят любое базисное решение этой системы, стараясь только найти его как можно проще и быстрее.

Любое найденное базисное решение необходимо проверить на допустимость, а любое допустимое решение следует проверить на оптимальность. Если допустимое решение не оптимально, то следует перейти к другому допустимому базисному решению.

Симплексный метод подразумевает, что при этом новом решении целевая функция приблизится к оптимальному решению, даже если не сумеет его достичь. Таким же образом проверяют каждое новое допустимое решение, пока не будет достигнуто оптимальное решение.

Найденное базисное решение может оказаться недопустимым. В таком случае с помощью симплексного метода осуществляют переход к другим базисным решениям, которые приближают нас к области допустимых решений. Процедуру повторяют пока на каком-то шаге решения не будет получено допустимое базисное решение и тогда можно применить алгоритм симплекс-метода, либо не убедимся в противоречивости системы ограничений задачи линейного программирования.

В процедуре симплекс-метода выделяют два основных этапа: первый этап состоит в получении допустимого базисного решения системы ограничений или установление факта ее несовместности; второй этап заключается в достижении оптимального решения. Каждый этап может состоять из несколько шагов (итераций), соответствующих получению того или иного базисного решения. Поскольку число базисных решений конечно, то число шагов симплекс-метода также является ограниченным.

Процедура симплекс-метода содержит четкие предписания о выполнении последовательных операций, что явно отражает его алго-

Процесс решения симплекс-методом был унифицирован, для его упрощения исходные данные задачи линейного программирования рекомендуется записывать в специальные симплекс-таблицы. Отсюда появилось название одной из модификаций симплекс-метода — *табличный симплекс-метод*. Задача линейного программирования, предназначенная для решения симплекс-методом, должна быть записана в *канонической* форме. Для приведения задачи к такому виду необходимо выполнить два условия:

- Задача линейного программирования, записанная в канонической форме, в общем случае имеет следующую структуру представления данных:

Здесь приняты обозначения $X = \{x_1, x_2, \dots x_n, \dots x_{n+m}\}$, $Z = Z(X)$ – целевая функция. Для краткости записи введем индексы для нумерации столбцов ($j = \overline{1, n+m}$) и строк ($i = \overline{1, m}$), свободные члены обозначим как $B = \{b_1, b_2, \dots b_m\}$. Так как при этом $Z_0 = 0$, то целевую функцию $Z(X)$ задачи линейного программирования можно записать в следующей компактной форме:

Рассмотрим для примера задачу на поиск максимума $Z(X)$.

15

— свободные переменные. Все переменные (базисные и свободные) разместим во второй строке таблицы. Базисные переменные дополнительно запишем в первый столбец таблицы и приведем к Жорданову базису, т.е. в столбце коэффициентов, соответствующих каждой базисной переменной, будет содержаться только одна плюс единица, а все остальные коэффициенты будут нулевыми. Например, в строке x_{n+1} столбца x_{n+1} записана плюс единица, а в остальных строках этого столбца записаны нули. Из первой строки таблицы возьмем коэффициенты целевой функции при базисных переменных и составим из них столбец c^B .

Таблица 1.3

Исходная симплекс-таблица

Базис	c^B	c_1	c_2	\dots	c_n	c_{n+1}	\dots	c_{n+m}	b_i	θ_i	Комментарий
		x_1	x_2	\dots	x_n	x_{n+1}	\dots	x_{n+m}			
x_{n+1}	c_{n+1}	$a_{1,1}$	$a_{1,2}$	\dots	$a_{1,n}$	1	\dots	0	b_1		
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots		
x_{n+m}	c_{n+m}	$a_{m,1}$	$a_{m,2}$	\dots	$a_{m,n}$	0	\dots	1	b_m		
$Z(X)$		Δ_1	Δ_2	\dots	Δ_n	0	\dots	0	Z_0		

Шаг 1. Для проверки на допустимость необходимо проверить на неотрицательность все элементы столбца свободных членов b_i . Если все они неотрицательны, то найдено *допустимое решение* и можно перейти к шагу 2. Если в столбце свободных членов найдены $b_i \leq 0$, то выбираем среди них наибольший по модулю – он задает ведущую строку (k). В этой строке (k) находим максимальный по модулю отрицательный элемент $a_{k,l}$ – он является ведущим элементом и задает ведущий столбец (l). Из базиса необходимо исключить переменную x_k (ведущая строка) и добавить новую переменную x_l (ведущий столбец). После этого необходимо пересчитать симплекс-таблицу согласно описанным выше правилам.

После пересчета симплекс-таблицы необходимо снова проверить на неотрицательность все элементы столбца свободных членов. Если таковых нет, то найденное решение допустимо и можно переходить ко второму шагу. Если среди них найдется хотя бы один отрицательный элемент (b_k), то в левой части соответствующей строки необходимо проверить все коэффициенты ($a_{k,i}$). Если все $a_{k,i} \geq 0$, то условия

задачи несовместны и задача не имеет решения. Если существует хотя бы один отрицательный коэффициент $a_{k,i}$, то снова переходим к первому шагу.

Шаг 2. Для того чтобы проверить решение на оптимальность, необходимо рассчитать элементы $\Delta_j = Z_j - c_j$ ($j = \overline{1, n}$) и разместить их в нижней индексной строке Z . При выполнении расчета будем полагать, что c^B — вектор-строка, A_j — вектор-столбец :

$$c^B = [c_{n+1}, c_{n+2}, \dots, c_{n+m}]; \quad A_j^T = [a_{j,n+1}, a_{j,n+2}, \dots, a_{j,n+m}].$$

Табл. 1.3 содержит значения всех элементов, которые необходимы для проверки на оптимальность решения путем расчета по формуле

$$\Delta_j = c^B \times A_j - c_j.$$

Из приведенной формулы следует, что для столбцов с базисными переменными $\Delta_{n+i} = 0$ ($i = \overline{1, m}$). Поэтому необходимо рассчитать лишь оценки Δ_j ($j = \overline{1, n}$) для столбцов небазисных переменных.

Если среди элементов симплекс-таблицы, находящихся в строке Z , нет отрицательных (текущее значение целевой функции Z_0 в расчет не принимаем), то для задачи на максимум найдено оптимальное решение.

Если в индексной строке Z есть отрицательные элементы то решение нуждается в улучшении. Среди отрицательных элементов индексной строки выберем максимальный по модулю элемент Δ_l (значение Z_0 не учитываем), тогда столбец (l) будет ведущим. Для нахождения ведущей строки, рассчитаем отношение соответствующего свободного члена и элемента из ведущего столбца, при условии, что они неотрицательны, т. е. $b_k > 0, a_{k,l} > 0$.

$$\theta_i = \frac{b_i}{a_{i,l}}, \quad i = \overline{1, m}.$$

Ведущей строкой будет строка ($i = k$) с минимальным значением θ_k . Элемент $a_{k,l}$ назовем ведущим (разрешающим). Из базиса исключается переменная x_k , соответствующая ведущей строке. Напротив, в базис включается переменная x_l , соответствующая ведущему столбцу. Все значения θ_i разместим в соответствующей колонке табл. 1.3.

Далее необходимо снова пересчитать симплекс-таблицу по формулам. Если после пересчета новой таблицы в строке Z остались отрицательные элементы, то необходимо снова перейти ко второму шагу.

Если в ведущем столбце нет положительных элементов, то невозможно найти ведущую строку. Это означает, что функция в области допустимых решений задачи не ограничена и алгоритм завершает свою работу.

Если в столбце свободных членов и в строке Z все элементы положительны, то найдено оптимальное решение (рис. 1.1).

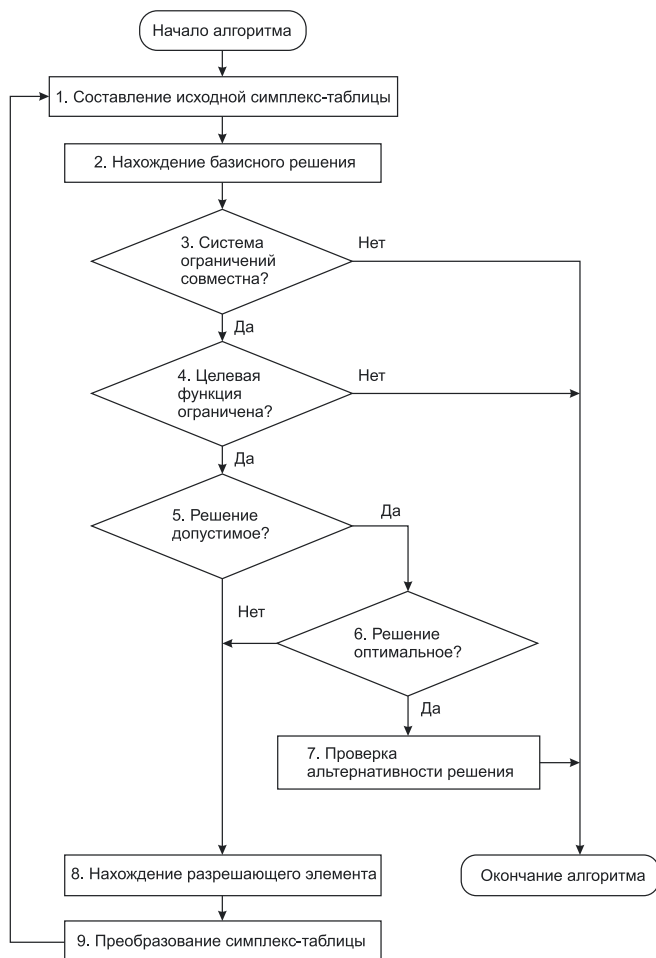


Рис. 1.1. Алгоритм симплекс-метода

Правила преобразований симплекс-таблицы. Сформулируем правила внесения изменений в новую симплекс-таблицу при преобразовании:

- по всей симплекс-таблице поменяем обозначения $\langle x_k \rangle \Leftrightarrow \langle x_l \rangle$;

скорректируем при них коэффициенты целевой функции « c_k » \Leftrightarrow « c_l »;

- на месте разрешающего элемента в новой таблице запишем $1/a_{k,l}$;
- все элементы ведущего столбца (кроме $a_{k,l}$) умножим на $-1/a_{k,l}$;
- все элементы ведущей строки (кроме $a_{k,l}$) и элемент b_l столбца свободных членов умножим на $1/a_{k,l}$; однако элементы $a_{k,j}$ в столбцах новых базисных переменных не подлежат преобразованию;
- все остальные элементы $a_{i,j}$ и b_i , не относящиеся к ведущей строке, ведущему столбцу и столбцам новых базисных переменных, пересчитываем по формуле прямоугольника

$$(a_{i,j})' = a_{i,j} - \frac{a_{i,l} \times a_{k,j}}{a_{k,l}}; \quad (b_i)' = b_i - \frac{a_{i,l} \times b_k}{a_{k,l}}.$$

Схема преобразования носит название *метод прямоугольника*, потому что преобразуемый элемент ($a_{i,j}$ или b_i) и соответствующие ему три сомножителя могут быть представлены как вершины *прямоугольника*.

1.3. Решение прямой и двойственной задач

1.3.1. Постановка задачи

Найти минимальное значение линейной функции

$$Z_{\min} = 10x_1 - 7x_2 - 5x_3 \quad \text{при ограничениях:}$$

$$\begin{cases} 6x_1 + 15x_2 + 6x_3 \leq 9, \\ 14x_1 + 42x_2 + 16x_3 \leq 21, \\ 2x_1 + 8x_2 + 2x_3 \leq 4. \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3);$$

Записать оптимальные планы исходной и двойственной задач.

1.3.2. Формулировки прямой и двойственной задач

Прямая задача (минимум). Поскольку в задаче на нахождение минимума все неравенства в системе ограничений-неравенств должны иметь вид « \geq », то необходимо умножить левую и правую части

первого неравенства в системе ограничений прямой задачи на «-1».

$$\begin{array}{cccc}
 & j = 1 & j = 2 & j = 3 & b_i \\
 c_1 = 10 & c_2 = -7 & c_3 = -5 & & \\
 i = 1 & -6x_1 & -15x_2 & -6x_3 & \geq -9, \\
 i = 2 & -14x_1 & -42x_2 & -16x_3 & \geq -21, \\
 i = 3 & -2x_1 & -8x_2 & -2x_3 & \geq -4, \\
 & x_j \geq 0 & (j = 1, 2, 3). & &
 \end{array}
 \quad Z = 10x_1 - 7x_2 - 5x_3.$$

В формулировке прямой задачи исходные матрицы имеют вид:

$$[a_{ij}] = \begin{bmatrix} -6 & -15 & -6 \\ -14 & -42 & -16 \\ -2 & -8 & -2 \end{bmatrix}; \quad [b_i] = \begin{bmatrix} -9 \\ -21 \\ -4 \end{bmatrix}; \quad [c_j] = [10 \quad -7 \quad -5].$$

Индексы будут изменяться в диапазонах $j = \overline{1, n}$; $i = \overline{1, m}$. Количество строк и столбцов матрицы $[a_{ij}]$ системы ограничений прямой задачи в рассматриваемом примере соответствует $m = 3$ и $n = 3$.

Двойственная задача (максимум). При формулировке двойственной задачи будем использовать матричные элементы прямой задачи. Кроме того, необходимо также учесть некоторые дополнительные правила преобразования:

1. Матрицу системы ограничений двойственной задачи получим путем транспонирования матрицы ограничений прямой задачи, т.е. $[a_{ij}]^T$.

2. В системе ограничений двойственной задачи вектор-столбец свободных членов представляет собой транспонированную вектор-строку целевой функции Z прямой задачи, т.е. $[c_j]^T$.

3. В целевой функции W двойственной задачи вектор-строка коэффициентов представляет собой транспонированный вектор-столбец свободных членов системы ограничений прямой задачи, т.е. $[b_i]^T$.

4. Если все неравенства исходной прямой задачи (минимум) были записаны в виде « \geq », то все неравенства двойственной задачи (максимум), соответствующие столбцам с $x_j \geq 0$, должны иметь вид « \leq ».

Тогда двойственную задачу сформулируем в следующем виде:

$$\begin{array}{rcccl}
 & i = 1 & i = 2 & i = 3 & c_j^T \\
 & b_1 = -9 & b_2 = -21 & b_3 = -4 & \\
 j = 1 & -6y_1 & -14y_2 & -2y_3 & \leq 10, \\
 j = 2 & -15y_1 & -42y_2 & -8y_3 & \leq -7, \\
 j = 3 & -6y_1 & -16y_2 & -2y_3 & \leq -5, \\
 & i = 1, 2, 3.
 \end{array} \quad W = -9y_1 - 21y_2 - 4y_3.$$

1.3.3. Симплекс-метод решения прямой задачи

Вначале запишем расширенный вариант системы ограничений прямой задачи, которая была использована на предыдущем этапе для формулировки двойственной задачи. Это необходимо для приведения прямой задачи к общепринятому каноническому виду.

$$\begin{cases} -6x_1 - 15x_2 - 6x_3 - x_4 - 0 - 0 = -9, \\ -14x_1 - 42x_2 - 16x_3 - 0 - x_5 - 0 = -21, \\ -2x_1 - 8x_2 - 2x_3 - 0 - 0 - x_6 = -4, \\ x_j \geq 0 \quad (j = \overline{1, 6}). \end{cases} \quad (1.4)$$

Умножив систему ограничений на (-1) , получим канонический вид прямой задачи: найти минимальное значение линейной функции

$$Z(X) = 10x_1 - 7x_2 - 5x_3 + 0x_4 + 0x_5 + 0x_6$$

при ограничениях в виде равенств

$$\begin{cases} 6x_1 + 15x_2 + 6x_3 + x_4 + 0 + 0 = 9, \\ 14x_1 + 42x_2 + 16x_3 + 0 + x_5 + 0 = 21, \\ 2x_1 + 8x_2 + 2x_3 + 0 + 0 + x_6 = 4, \\ x_j \geq 0 \quad (j = \overline{1, 6}). \end{cases}$$

Получившуюся расширенную матрицу запишем в 1-й блок таблицы Гаусса. Столбец c^B (табл. 1.4) будет содержать коэффициенты целевой функции при базисных переменных:

Таблица 1.4

Первоначальный (нулевой) вариант

Базис	c^B	10	-7	-5	0	0	0	b_i	θ_i	Комментарий
		x_1	x_2	x_3	x_4	x_5	x_6			
x_4	0	6	15	6	1	0	0	9	$\frac{9}{15} = 0.6$	
x_5	0	14	42	16	0	1	0	21	$\frac{21}{42} = 0.5$	$\times (-\frac{15}{42})_1 (-\frac{8}{42})_3$
x_6	0	2	8	2	0	0	1	4	$\frac{4}{8} = 0.5$	
$Z(X)$		-10	7	5	0	0	0	0		

Шаг 1. Этот шаг состоит в проверке полученного решения на допустимость. Для проверки решения на допустимость необходимо проверить на положительность все элементы столбца свободных членов b_i . Если среди элементов столбца свободных членов нет ни одного отрицательного, то найденное решение считается допустимым и можно переходить ко второму шагу. Если столбец свободных членов содержит отрицательные элементы то среди них выберем максимальный по модулю — он задает ведущую строку (k). Среди элементов ведущей строки k найдем максимальный по модулю отрицательный элемент (a_{kl}), который называется ведущим элементом и задает ведущий столбец (l). Переменная, соответствующая ведущему столбцу, *добавляется* в базис. Напротив, переменная, соответствующая ведущей строке, из базиса *исключается*. После этого необходимо заново пересчитать всю симплекс-таблицу.

Если в столбце свободных членов найдены отрицательные элементы, а в соответствующей строке все элементы неотрицательны, то условия задачи считаются несовместными и у данной задачи нет решений.

После перерасчета таблицы в столбце свободных членов должны оставаться только неотрицательные элементы, тогда можно перейти ко второму шагу. В противном случае придется вернуться к первому шагу.

Шаг 2. На первоначальном этапе в качестве базисных переменных были заданы переменные x_4 , x_5 , x_6 , тогда начальное опорное решение:

$$x^{(0)} = (0; 0; 0; 9; 21; 4);$$

$$Z^{(0)} = 10 \times 0 - 7 \times 0 - 5 \times 0 + 0 \times 9 + 0 \times 21 + 0 \times 4 = 0.$$

Полученный опорный план требует проверки на оптимальность. Для проверки на оптимальность вычислим индексы $\Delta_j = Z_j - c_j$, которые разместим в нижней индексной строке табл. 1.4

$$\Delta_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 6 \\ 14 \\ 2 \end{bmatrix} - 10 = -10; \quad \Delta_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 15 \\ 42 \\ 8 \end{bmatrix} - (-7) = 7;$$

$$\Delta_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 6 \\ 16 \\ 2 \end{bmatrix} - (-5) = 5.$$

При решении задачи на минимум возможны три варианта:

- полученный план оптимален, поскольку индексная строка не содержит положительных оценок (элемент b_0 в расчет не берем);
- полученный план не оптимален и возможно его дальнейшее улучшение, поскольку есть хотя бы одна положительная оценка, а в столбце над ней есть хотя бы один положительный коэффициент;
- оптимального плана не существует и целевая функция не ограничена сверху, поскольку есть хотя бы одна положительная оценка, в столбце над которой нет ни одного положительного коэффициента.

В строке индексов нашей задачи есть положительные оценки, а в столбцах над ними есть положительные элементы. Опорный план не оптимален и допускает улучшение путем перехода к новому базису.

В случае задачи на минимум ведущий столбец (l) определим по наибольшей оценке Δ_j в индексной строке. Он указывает, какая переменная будет добавлена в новый базис. В нашем примере ведущий столбец $l = 2$ и переменная x_2 должна быть добавлена в новый базис.

Ведущую строку (k) определим по наименьшей величине θ_i . Она указывает, какая переменная будет выведена из базиса. В нашем примере ведущая строка $k = 2$ ($21:42=0.5$), поэтому переменная x_5 должна быть выведена из базиса. При выборе между строками с одинаковыми отношениями θ_i в качестве ведущей выбрали строку с меньшим номером.

Замечания к алгоритму преобразования матрицы:

- если $a_{il} \leq 0$, то θ_i не вычисляют;
- иногда при вычислении θ_i получается так, что минимум отношения оказывается одинаковым сразу для нескольких номеров i , т.е. сразу несколько строк таблицы претендуют на роль разрешающей строки. Произвольный выбор ведущей строки может привести к заикливанию алгоритма симплекс-метода (вырожденный случай). Чтобы избежать этой ситуации, выбор рекомендуют осуществлять по определенному правилу. Очень часто используют следующие правила:

а) в качестве разрешающей следует всегда выбирать строку с наименьшим номером в системе ограничений;

б) для строк таблицы с одинаковым минимумом отношений θ_i вычисляются отношения a_{i1}/a_{i1} , и находится строка, для которой это отношение является минимальным. Если такая строка единственная, то ее считают разрешающей. В противном случае вычисляются следующие отношения a_{i2}/a_{i1} и т. д. В результате такого перебора должна остаться единственная разрешающая строка.

В данной задаче были определены ведущий столбец $l = 2$ и ведущая строка $k = 2$. Теперь необходимо перейти к новому базису и составить для него симплекс-таблицу. Для перехода от одного блока таблицы к другому используем элементарные преобразования Гаусса для строк.

После определения разрешающего элемента следует добиваться, чтобы столбец с этим разрешающим элементом стал единичным, т. е. все элементы в нём стали 0, а сам разрешающий элемент стал 1.

Для того чтобы в первой строке вместо 15 получить 0 при её сложении со второй строкой, перед сложением вторую строку умножим на

$$\frac{1}{42}(-15) = -\frac{5}{14}.$$

Для получения в 3-й строке нуля вместо 8 при сложении её со 2-й строкой, 2-ю строку перед сложением надо умножить на

$$\frac{1}{42}(-8) = -\frac{4}{21}.$$

Таким образом, определены коэффициенты преобразования, которые зафиксируем в крайней правой колонке «комментарии», табл. 1.5.

Сложим по очереди все строки со второй строкой и получим нули в разрешающем столбце. Однако сам разрешающий элемент пока неизменен. Разделим вторую строку на 42, столбец станет равен 1, табл. 1.5.

В результате проведенного преобразования старой симплекс-таблицы в новой симплекс-таблице вместо старого ведущего столбца получим единичный столбец. Теперь можно построить новый опорный план.

$$\begin{aligned} x^{(1)} &= (0; \frac{1}{2}; 0; \frac{3}{2}; 0; 0); \\ Z^{(1)} &= 10 \times 0 - 7 \times \frac{1}{2} - 5 \times 0 + 0 \times \frac{3}{2} + 0 \times 0 + 0 \times 0 = -\frac{7}{2}. \end{aligned}$$

Опорный план проверим на оптимальность путем вычисления значений индексов $\Delta_j = Z_j - c_j$, которые разместим в нижней индексной строке табл. 1.5:

Таблица 1.5

Первый вариант

Базис	c^B	10	-7	-5	0	0	0	b_i	θ_i	Комментарий
		x_1	x_2	x_3	x_4	x_5	x_6			
x_4	0	1	0	2/7	1	-5/14	0	3/2	$\frac{3/2}{2/7} = 5.3$	
x_2	-7	1/3	1	8/21	0	1/42	0	1/2	$\frac{1/2}{8/21} = 1.3$	$(-\frac{3}{4})_1 (\frac{11}{4})_3$
x_6	0	-2/3	0	-22/21	0	-4/21	1	0		
$Z(X)$		-37/3	0	7/3	0	-1/6	0	-7/2		

$$\Delta_1 = \begin{bmatrix} 0 \\ -7 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 1 \\ 1/3 \\ -2/3 \end{bmatrix} - 10 = -\frac{37}{3}; \quad \Delta_3 = \begin{bmatrix} 0 \\ -7 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 2/7 \\ 8/21 \\ -22/21 \end{bmatrix} - (-5) = \frac{7}{3};$$

$$\Delta_5 = \begin{bmatrix} 0 \\ -7 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} -5/14 \\ 1/42 \\ -4/21 \end{bmatrix} - 0 = -\frac{1}{6}.$$

Опорный план (табл. 1.5) не оптимален, т.к. в нем есть положительный индекс. Необходимо перейти к новому опорному плану, представленному в табл. 1.6.

Таблица 1.6

Второй вариант

Базис	c^B	10	-7	-5	0	0	0	b_i	θ_i	Комментарий
		x_1	x_2	x_3	x_4	x_5	x_6			
x_4	0	3/4	-3/4	0	1	-3/8	0	9/8		
x_3	-5	7/8	21/8	1	0	1/16	0	21/16		
x_6	0	1/4	11/4	0	0	-1/8	1	11/8		
$Z(X)$		-115/8	-49/8	0	0	-5/16	0	-105/16		

$$x^{(2)} = (0; 0; \frac{21}{16}; \frac{9}{8}; 0; \frac{11}{8});$$

$$\begin{aligned} Z^{(2)} &= 10 \times 0 - 7 \times 0 - 5 \times \frac{21}{16} + 0 \times \frac{9}{8} + 0 \times 0 + 0 \times \frac{11}{8} = \\ &= -\frac{105}{16} = -6 \frac{9}{16}. \end{aligned}$$

Проверим новый опорный план на оптимальность: вычислим значения индексов $\Delta_j = Z_j - c_j$, которые поместим в нижнюю индексную строку табл. 1.6

$$\Delta_1 = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 3/4 \\ 7/8 \\ 1/4 \end{bmatrix} - 10 = -\frac{115}{8}; \quad \Delta_2 = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} -3/4 \\ 21/8 \\ 11/4 \end{bmatrix} - (-7) = -\frac{49}{8};$$

$$\Delta_5 = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} -3/8 \\ 1/16 \\ -1/8 \end{bmatrix} - 0 = -\frac{5}{16}.$$

Все оценки в строке индексов неположительны, поэтому опорный план, составленный по последней симплекс-таблице, является оптимальным. Найденное решение является оптимальным с точки зрения достижения минимума целевой функции при заданных ограничениях:

$$x_1 = 0; \quad x_2 = 0; \quad x_3 = \frac{21}{16}.$$

Минимально возможное значение целевой функции при этом будет

$$Z(0; 0; \frac{21}{16}) = -\frac{105}{16}.$$

1.3.4. Оптимальное решение двойственной задачи

В третьей теореме двойственности доказано $\bar{Y} = C^B \times D^{-1}$. Вектор-строку $C^B = [0, -5, 0]$ возьмем из соответствующей колонки табл. 1.6. Матрицу D получим по следующей технологии: из последней итерации прямой задачи запишем оптимальный базис $[x_4, x_3, x_6]$, однако компоненты векторов $[x_4, x_3, x_6]$ возьмем из исходной расширенной матрицы (1.4). Составим матрицу D из компонентов векторов $[x_4, x_3, x_6]$ и любым доступным способом определим обратную матрицу D^{-1} .

$$D = [x_4, x_3, x_6] = \begin{bmatrix} -1 & -6 & 0 \\ 0 & -16 & 0 \\ 0 & -2 & -1 \end{bmatrix}; \quad D^{-1} = \begin{bmatrix} -1 & 3/8 & 0 \\ 0 & -1/16 & 0 \\ 0 & 1/8 & -1 \end{bmatrix}.$$

При выполнении этих требований оптимальное решение двойственной задачи ($\bar{Y} \equiv [y_1, y_2, y_3]$) будет определено следующим образом:

$$\bar{Y} = C^B \times D^{-1} = [0, -5, 0] \times \begin{bmatrix} -1 & 3/8 & 0 \\ 0 & -1/16 & 0 \\ 0 & 1/8 & -1 \end{bmatrix} = [0, \frac{5}{16}, 0].$$

Соответствующее ему оптимальное значение целевой функции будет

$$W_{\max} = -9 \cdot 0 - 21 \cdot \frac{5}{16} - 4 \cdot 0 = -\frac{105}{16} = Z_{\min}.$$

1.4. Решение задач оптимизации в среде Матлаб

1.4.1. Функция linprog

Среда Матлаб содержит встроенную функцию `linprog`, предназначенную для решения задач линейного программирования, которая решает задачу линейного программирования, представленную в форме:

$$\begin{aligned} Z &= C \times X \rightarrow \min, \\ A \times X &\leq B, \\ A_{eq} \times X &= B_{eq}, \\ LB &\leq X \leq UB. \end{aligned} \tag{1.5}$$

Условные обозначения, использованные для входных данных:

- C — вектор-строка целевой функции;
- A — матрица ограничений-неравенств;
- B — вектор-строка правых частей ограничений-неравенств;
- Aeq — матрица ограничений-равенств;
- Beq — вектор правых частей ограничений-равенств;
- LB — вектор-столбец, ограничивающий план X снизу;
- LU — вектор-столбец, ограничивающий план X сверху.

Результат работы `linprog` — оптимальный план задачи (1.5) в виде вектор-столбца X и экстремальное значение целевой функции Z_{\min} .

Для иллюстрации возьмем модельную задачу предыдущего примера, который был ранее решен вручную. Исходные данные, необходимые для вызова процедуры, подготовим следующим образом (рис. 1.2):

Строка 0. Заголовок функции, реализующей процедуру.

Строки 2–5. Подготовка исходных данных в матричном виде.

```
%.....
00 function LinearProgram1()
01 % Исходные данные
02 C = [10 -7 -5]; % Коэффициенты целевой функции
03 A = [6 15 6; 14 42 16; 2 8 2]; % Строки м-цы ограничений-неравенств
04 B = [9 21 4]; % Вектор правых частей ограничений-неравенств
05 LB = [0; 0; 0]; % Вектор, ограничивающий план x снизу
06 % Решение задачи
07 [xopt,zmin] = linprog(C,A,B,[],[],LB);
08 % Вывод данных на экран
09 Out='x1 = %1.5f\nx2 = %1.5f\nx3 = %1.5f\nZmin(x1,x2,x3)=%1.5f;\n';
10 fprintf(1,Out,xopt,zmin);
11 end %.....
```

Рис. 1.2. Листинг первой программы в *m*-кодах Матлаб

Строка 7. Вызов процедуры `linprog` и получение результатов расчета. В списке аргументов функции `linprog` квадратные скобки означают, что данный аргумент не используется в текущем расчете. В данной задаче не были использованы два аргумента, соответствующие строкам матрицы ограничений-равенств `Aeq` и правым частям ограничений-равенств `Beq`. Пропуск аргумента в конце списка не требует его указания квадратными скобками. В текущей задаче в конце списка пропущен аргумент `UB`, поскольку в задаче не предусмотрено ограничение сверху.

Строки 9, 10. В этих строках программы осуществляется форматный вывод результатов расчета на экран панели Матлаб.

Параметр `exitflag` позволяет выводить информацию о том, как завершилось решение задачи Матлаб. Если значение параметра `exitflag` равно единице, то найдено решение задачи, если равно 0, то превышено допустимое число итераций, если равно -2, то множество планов задачи пусто, если равно -3, то целевая функция не ограничена снизу на множестве планов. Для симплекс-метода допустимое число итераций по умолчанию в десять раз больше количества переменных.

Выходной параметр `output` содержит информацию о процессе оптимизации, в частности, число итераций `output.iterations` и используемый алгоритм `output.algorithm`.

Выходной параметр `lambda` содержит результат решения двойственной задачи линейного программирования.

Параметр `lambda` состоит из нескольких массивов: `lambda.ineqlin`, `lambda.eqlin`, `lambda.upper`, `lambda.lower`. В этих массивах находятся двойственные переменные, приписанные ограничениям-неравенствам, ограничениям-равенствам, ограничениям на план сверху и снизу соответственно. На рис. 1.3 представлен вариант программы, реализую-

щий расширенный вывод результатов расчета рассматриваемой задачи на экран панели Матлаб.

```
00 function LinearProgram2() %.....
01 % Исходные данные
02 C = [10 -7 -5]; % Коэффициенты целевой функции
03 A = [6 15 6; 14 42 16; 2 8 2]; %Строки м-цы ограничений-неравенств
04 B = [9 21 4]; % Вектор правых частей ограничений-неравенств
05 LB = [0; 0; 0]; %Вектор, ограничивающий план x снизу
06 % Решение задачи
07 [xopt,zmin,exitflag,output,lambda] = linprog(C,A,B,[],[],LB);
08 % Вывод данных на экран
09 Out='x1 = %1.5f\nx2 = %1.5f\nx3 = %1.5f\nZmin(x1,x2,x3)=%1.5f;\n';
10 fprintf(1,Out,xopt,zmin);
11 fprintf(1,'ExitFlag=%d\nIterats=%d\n',exitflag,output.iterations);
12 fprintf(1,'y1 = %1.5f\ny2 = %1.5f\ny3 = %1.5f\n',lambda.ineqlin);
13 end %.....
```

Рис. 1.3. Листинг второй программы в *m*-кодах Матлаб

Ниже представлена распечатка результатов расчета текущей задачи, включающая оптимальный план $\bar{X} = [x_1, x_2, x_3]$, значение целевой функции Z_{\min} для оптимального плана и $\bar{Y} = [y_1, y_2, y_3]$.

```
>> LinearProgram2
Optimization terminated.
x1 = 0.00000
x2 = 0.00000
x3 = 1.31250
Zmin(x1,x2,x3) = -6.56250;
ExitFlag=1
Iterats=8
y1 = 0.00000
y2 = 0.31250
y3 = 0.00000
>>
```

Ручной расчет той же самой задачи был выполнен нами ранее при помощи табличного симплекс-метода. Приведем ещё раз для сравнения основные результаты, которые были получены при ручном расчете.

$$\bar{X} = [0, 0, \frac{21}{16}], \quad Z_{\min} = -\frac{105}{6}, \quad \bar{Y} = [0, \frac{5}{16}, 0].$$

Из сравнения приведенных результатов результатов ручного расчета с результатами машинного расчета с использованием функции `linprog` видно их совпадение. Однако данные машинного расчета приведены в виде округленных чисел, а данные ручного расчета представлены в виде простых дробей, т. е. не содержат ошибки округления.

Программа `linprog` поддерживает несколько различных режимов оптимизации. При сравнении результатов ручного расчета табличным симплекс-методом и результатов машинного расчета рекомендуется принудительно задать режим использования только симплекс-метода и запретить использование альтернативных режимов оптимизации. При наличии нескольких оптимальных планов решаемой задачи такой режим повышает вероятность совпадения оптимальных планов машинного и ручного расчетов. Программа `linprog` имеет дополнительные возможности для задания такого режима работы, рис. 1.4.

```
opt = optimset('LargeScale','off','Simplex','on');
[Xo,Zm,exitflag,output,lambda] = linprog(C,A,B,[],[],LB,[],[],opt);
```

Рис. 1.4. Фрагмент программы в m -кодах Матлаб, необходимый для задания режима использования только симплекс-метода и запрета использования альтернативных режимов оптимизации

1.5. О недостатках симплексного метода

Запишем задачу линейного программирования в стандартизованном каноническом виде

$$\begin{aligned} (c, x) &\rightarrow \min, \\ \hat{A}x &= b, \quad x \geq 0. \end{aligned} \tag{1.6}$$

Здесь использованы обозначения $x \in R^m$, $b \in R^n$; R^m , R^n для евклидовых пространств соответствующих размерностей. Для исключения возможной неоднозначности в решении задачи (1.6), под решением задачи будем понимать нормальное решение (решение с минимальной нормой). Очевидно, что для построения приближенного решения (1.6), непрерывно зависящего от входных данных A , c , b , необходимо использовать какой-либо регуляризирующий алгоритм. Как показывает приводимый ниже простой пример, наиболее употребительный «точный» метод численного решения (1.6) – симплекс-метод – этим свойством не обладает. Пусть

$$A = \begin{bmatrix} 2 & -2 & 1 & 1 & 1/2 \\ 1 & 1 & 1 & 0 & 1/2 \\ 3 & -1 & 2 & 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \quad c = (1, 1, 0, 0, 0). \tag{1.7}$$

Задача (1.6) имеет нормальное решение $x^* = (0; 0; 0.8; 1; 0.4)$. Стандартная программа решения задач линейного программирования на основе симплекс-метода при точных данных выдала в качестве решения задачи (1.6) вектор $(0; 0; 1; 1; 0)$. Этот вектор также является решением задачи (1.6), однако он не обладает минимальной нормой.

Внесем возмущение в условия задачи (1.6), положив $b_3 = 3.01$ (вместо 3.0). Все остальные исходные данные оставим без изменений. Стандартная программа не выдает никакого решения и сообщает о несовместности ограничений. Такой же ответ получался при других возмущениях (даже очень малых). Все они делают исходную задачу несовместной.

С точки зрения классического подхода эти результаты вполне объяснимы: они свидетельствуют о высоком качестве стандартной программы. Напротив, с точки зрения теории регуляризации эти результаты свидетельствуют о непригодности симплекс-метода в качестве регуляризирующего алгоритма. В таком случае для решения задачи линейного программирования (1.7) необходимо использовать какие-либо итерационные методы с гарантированными регуляризирующими свойствами.

1.6. Выполнение численного расчета

1. Получив индивидуальное задание, нужно сформулировать прямую задачу в рекомендуемом (нерасширенном) виде: записать целевую функцию прямой задачи и систему ограничений (ограничения-неравенства в рекомендуемом виде и/или ограничения-равенства). Заполнить исходную (нерасширенную) симплекс-таблицу прямой задачи.

2. Сформулировать двойственную задачу в рекомендуемом (нерасширенном) виде: записать целевую функцию двойственной задачи и систему ограничений (ограничения-неравенства в рекомендуемом виде и/или ограничения-равенства).

3. Выбрать одну из задач (прямая или двойственная) для процедуры оптимизации. Записать исходную систему ограничений-равенств и заполнить исходную (расширенную) симплекс-таблицу.

4. Осуществить процедуру оптимизации табличным симплекс-методом и получить оптимальный план и значение целевой функции в найденной точке экстремума.

5. По теореме двойственности рассчитать для сопряженной задачи оптимальный план и значение целевой функции двойственной задачи в найденной точке экстремума.

6. Подготовить исходные данные для машинного расчета и выполнить расчет в среде Матлаб с помощью встроенной процедуры `linprog`. Провести анализ и сопоставление результатов машинного расчета с результатами ручного расчета.

7. Повторить п.6 после внесения небольших (около 0.1–1 %) изменений в какой-либо свободный член (ограничения типа равенств) прямой задачи. Сопоставить результаты с п.6. и сделать выводы.

1.7. Упражнения для самоконтроля

1. Используя свойство симметричности, с помощью двойственной задачи решить исходную. Найти минимальное значение линейной функции

$$Z = 4x_1 + 5x_2 + 6x_3$$

при ограничениях

$$\begin{cases} x_1 + x_2 + x_3 \geq 5, \\ x_1 - x_2 + 2x_3 \geq 1, \\ x_1 - x_2 - 4x_3 \leq -3, \\ x_1 - x_2 + 8x_3 \geq 4, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

2. Для двойственной задачи записать исходную и найти ее решение:

$$Z_{\min} = 2y_1 + 4y_2 + 12y_4$$

при ограничениях

$$\begin{cases} y_1 + 2y_2 + y_3 + 4y_4 \geq 10, \\ 2y_1 + y_2 - 2y_3 + 3y_4 \geq 4, \end{cases} \quad y_i \geq 0 \quad (i = 1, 2, 3, 4).$$

Записать оптимальные планы исходной и двойственной задач.

3. Применяя двойственный симплексный метод, решить задачу

$$Z_{\max} = 2x_1 - 2x_2 + 3x_3 + 4x_4$$

при ограничениях

$$\begin{cases} x_1 + 2x_2 + x_3 + x_4 \leq 2, \\ 2x_1 - x_2 + 2x_3 - 3x_4 \geq 3, \\ 3x_1 + 4x_2 - 5x_3 + 2x_4 \leq 4, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3, 4).$$

Записать оптимальные планы исходной и двойственной задач.

2. Регуляризация на компактных множествах

Использование дополнительной (*априорной*) информации об искомом решении является основой многих численных алгоритмов приближенных решений некорректно поставленных задач, устойчивых к малым изменениям исходных данных. Наличие дополнительной информации количественного характера позволяет ограничить класс возможных решений только, например, *компактным множеством*, и задача становится устойчивой к малым изменениям исходных данных. Иногда для нахождения приближенных решений бывает достаточно лишь качественной информации о решении, в таком случае различают методы *подбора*, *квазиразрешения* и некоторые другие методы.

2.1. Понятие компактных пространств

Компактным множеством (компактом) называют множество, из любой последовательности элементов которого можно выделить подпоследовательность, сходящуюся к некоторому его элементу [2]. Понятие компакта часто применяют к метрическим пространствам.

Компактное пространство обобщает свойства *ограниченности* и *замкнутости* евклидовых пространств на произвольное топологическое пространство. В общей топологии компактные пространства по своим свойствам играют такую же роль, как конечные множества в теории множеств.

По определению А. Н. Колмогорова [6]: топологическое пространство называется *компактным*, если всякое покрытие этого пространства открытыми множествами содержит конечное подпокрытие.

Компактные множества обладают целым рядом практически полезных свойств, приведем без доказательства лишь некоторые из них:

- 1) в метрическом пространстве компактное множество замкнуто;
- 2) в метрическом пространстве компактное множество ограничено;
- 3) в евклидовом пространстве любое замкнутое и ограниченное множество является компактным;
- 4) если непрерывное отображение $f : X \rightarrow f(X)$, то из компактности пространства X следует компактность множества $f(X)$. Иными словами, непрерывный образ компактного пространства также компактен.

2.1.1. Некорректные задачи на компактах

1. Вводные понятия. Рассмотрим операторное уравнение:

$$\hat{\mathbf{A}}z = u, \quad (2.1)$$

где $\hat{\mathbf{A}}$ – инъективный линейный оператор, действующий из нормированного пространства Z в нормированное пространство U . Пусть \bar{z} – точное решение операторного уравнения, $\hat{\mathbf{A}}\bar{z} = \bar{u}$, \bar{u} – точная правая часть. Зададим приближенную правую часть u_δ такую, что $\|\bar{u} - u_\delta\| \leq \delta$, $\delta > 0$. Отображение схематически показано на рис. 2.1, а.

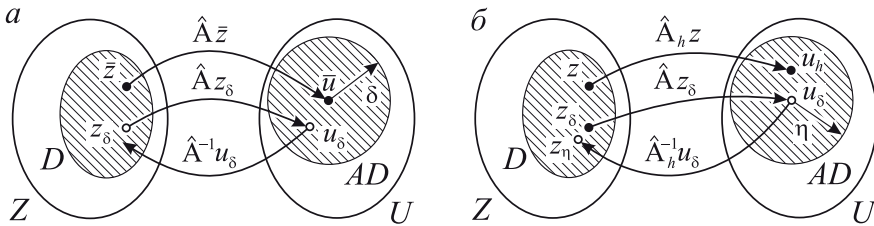


Рис. 2.1. Схема метода регуляризации на компактных множествах при наличии погрешностей: а – только правой части; б – правой части и оператора

Множеством приближенных решений операторного уравнения называют множество $Z_\delta = \{z_\delta : \|\hat{\mathbf{A}}z_\delta - u_\delta\| \leq \delta\}$. Обратный оператор $\hat{\mathbf{A}}^{-1}$ линейной некорректной задачи неограничен, поэтому для любого $\delta > 0$ имеем $\text{diam } Z_\delta = \sup\{\|z_1 - z_2\| : z_1, z_2 \in Z_\delta\} = \infty$.

Для приближенного решения уравнения (2.1) в вычислительной практике широко используется т.н. *метод подбора*. Этот метод состоит в том, что решают прямую задачу для элементов z , которые принадлежат некоторому заранее заданному подклассу возможных решений D ($D \subset Z$), т.е. вычисляется оператор $\hat{\mathbf{A}}z$. При этом в качестве приближенного решения принимают любой элемент z_0 из множества D , на котором невязка $\rho_U(\hat{\mathbf{A}}z, u)$ достигает минимума.

2. Постановка задачи. Успешное практическое применение метода подбора потребовало его теоретического обоснования. Для ограничения класса возможных решений D , при которых метод подбора является устойчивым и $z_n \rightarrow \bar{z}$, одним из общекритериальных требований является *компактность множества D*.

Теорема 4 [7, с. 39–40]. Пусть инъективный непрерывный оператор $\hat{\mathbf{A}}: D \in Z \rightarrow AD \in U$, где Z, U – нормированные пространства, D – компакт. Тогда оператор $\hat{\mathbf{A}}^{-1}$ непрерывен на AD .

А. Н. Тихонов применил теорему 4 к доказательству утверждения: если множество решений является *компактом* и приближенная правая часть операторного уравнения принадлежит образу этого компакта, то задача решения операторного уравнения будет корректной.

Задача, корректная по Тихонову. Если для точного значения \bar{u} существует единственное решение \bar{z} уравнения (2.1), принадлежащее заданному компакту D , то задача (2.1) называется корректной по Тихонову, а обратный оператор \hat{A}^{-1} будет непрерывен на множестве AD . Если вместо элемента \bar{u} нам известен u_δ такой, что $\rho_U(\bar{u}, u_\delta) \leq \delta$ и $u_\delta \in AD$, то за приближенное решение уравнения (2.1) с правой частью $u = u_\delta$ можно взять элемент $z_\delta = \hat{A}^{-1} u_\delta$. Если $u_\delta \in AD$, то z_δ будет стремиться к \bar{z} при $\delta \rightarrow 0$. Множество возможных решений, сопоставимых с искомым по точности, на котором задача нахождения уравнения (2.1) является корректно поставленной, называется *классом корректности* (*множеством корректности*). Если инъективный оператор \hat{A} непрерывен, то компакт D , к которому принадлежит \bar{z} , является классом корректности для уравнения (2.1). Схематично это показано на рис. 2.2, а.

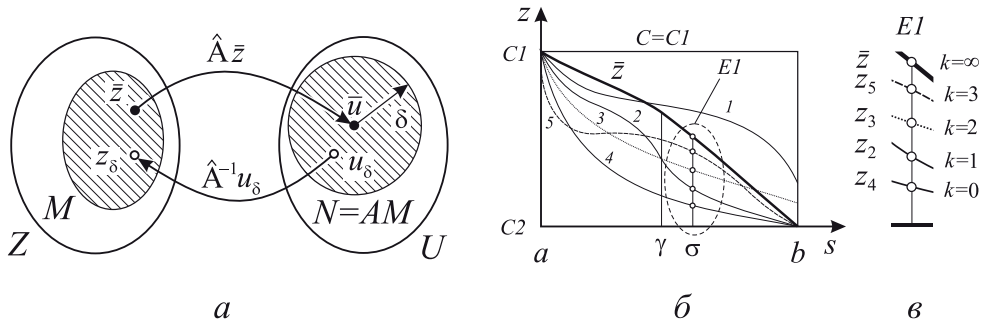


Рис. 2.2. Схематичное пояснение метода регуляризации на компактных множествах: а – задача, корректная по Тихонову; б – регуляризация на множестве монотонных функций; в – формирование последовательности $E1$

Любой элемент $z_\delta \in D$: $z_\delta = \arg \min_{z \in D} \|\hat{A}z - u_\delta\|$ будет называться *квазирешением* уравнения (2.1) на компакте D . Здесь выражение $z_\delta = \arg \min_{z \in D} \|\hat{A}z - u_\delta\| \Leftrightarrow \|\hat{A}z_\delta - u_\delta\| = \min \{\|\hat{A}z - u_\delta\|: z \in D\}$.

Несмотря на то, что из существования квазирешения вовсе не следует его единственность, тем не менее любое квазирешение при $\delta \rightarrow 0$ сходится к точному решению, т. е. $z_\delta \rightarrow \bar{z}$. Для осуществления этой сходимости не обязательно знать погрешность δ . Однако если погрешность δ все же известна, то в качестве приближенного решения

(квазирешения) может быть выбран любой элемент $z_\delta \in D$, удовлетворяющий неравенству: $\|\hat{\mathbf{A}}z_\delta - u_\delta\| \leq \delta$. Более того, можно найти погрешность приближенного решения, решив экстремальную задачу на нахождение $\max \|z - z_\delta\|$ по всем $z \in D$, удовлетворяющим неравенству: $\|\hat{\mathbf{A}}z - u_\delta\| \leq \delta$. Отметим, что точное решение \tilde{z} также должно удовлетворять данному неравенству.

Резюмируя сказанное, отметим, что задача отыскания квазирешения лишь незначительно отличается от корректной задачи, а именно, здесь не выполняется только условие единственности квазирешения.

3. Оператор с погрешностью. Для оператора $\hat{\mathbf{A}}$, заданного с известной погрешностью h , понятие квазирешения можно определить аналогичным образом, если заменить этот оператор $\hat{\mathbf{A}}$ на соответствующий оператор с погрешностью $\hat{\mathbf{A}}_h$ (рис. 2.1, б).

В этом случае квазирешением операторного уравнения (2.1) на компакте D называется элемент $z_\eta \in D$: $z_\eta = \arg \min_{z \in D} \|\hat{\mathbf{A}}_h z - u_\delta\|$.

В качестве квазирешения (приближенного решения) можно выбрать любой элемент $z_\eta \in D$, который будет удовлетворять неравенству:

$$\|\hat{\mathbf{A}}z_\eta - u_\delta\| \leq \delta + h \|z_\eta\|.$$

Функционал невязки $\|\hat{\mathbf{A}}z - u_\delta\|^2$, определенный в гильбертовых пространствах Z и U , является выпуклым и дифференцируемым. На этом свойстве функционала основаны многие численные методы отыскания квазирешений для линейных операторных уравнений. Пусть D – выпуклый компакт, тогда для нахождения квазирешения необходимо решить задачу выпуклого программирования. В качестве критериев остановки процесса минимизации функционала невязки можно использовать те же самые неравенства, которые были записаны выше для определения приближенных решений на компактах. При отыскании погрешности найденного приближенного решения требуется найти максимум выпуклого функционала, в то время как стандартная задача выпуклого программирования предполагает поиск минимума целевого функционала.

Множества корректности возникают при решении прикладных задач, где используют априорную информацию о принадлежности точного решения конечно-параметрическому семейству функций. Благодаря этому задача отыскания параметров может оставаться корректной, даже если задача без априорной информации не является корректно поставленной задачей.

Пусть решение операторного уравнения принадлежит классу функций, о котором известно, что функции $z(s)$, $s \in [a, b]$ являются «монотонными и ограниченными» или «монотонными и выпуклыми», или «ограниченными и выпуклыми», или «имеют заданное число максимумов и минимумов». Этой априорной информации оказывается достаточно для выделения компакта в пространстве $L_2[a, b]$. После перехода к конечно-разностной аппроксимации для отыскания квазирешения можно применить известные методы квадратичного программирования, например, метод проекции сопряженных градиентов или метод условного градиента. При использовании численных методов поиска квазирешения возможно также отыскание погрешности приближенного решения.

2.2. Свойства метода регуляризации на компактных множествах

При решении многих обратных задач математической физики бывает доступна качественная информация о характере искомого решения: *монотонность, выпуклость, монотонность и выпуклость, ограниченность вариации*. Эта априорная информация носит достаточно общий характер, но часто ее оказывается вполне достаточно для построения устойчивых алгоритмов приближенного решения уравнения (2.1). Рассмотрим четыре упомянутых случая более подробно.

1. Множество монотонных функций. В этом случае из априорной информации известно, что для некорректно поставленной задачи (2.1) существует точное решение $\bar{z}(s)$, определенное на отрезке $s \in [a, b]$ и являющееся монотонной функцией (без потери общности считаем ее невозрастающей), которая ограничена сверху и снизу соответственно константами $C_1 = C > 0$, $C_2 = 0$. Пусть $Z \downarrow_C$ – множество невозрастающих функций $z(s)$, ограниченных константами C и 0 , т. е. $0 \leq z(s) \leq C$ для $\forall s \in [a, b]$ (рис. 2.2, б); $\{z_1(s), z_2(s), \dots, z_n(s), \dots\}$ – последовательность функций из $Z \downarrow_C$. Тогда по теореме Хелли о выборе [6, с. 86–88] существуют подпоследовательность

$$E1 \equiv \{z_{n_1}(s), z_{n_2}(s), \dots, z_{n_k}(s), \dots\},$$

функция $\bar{z}(s) \in Z \downarrow_C$ и последовательность индексов n_k такие, что в каждой точке $s \in [a, b]$

$$\lim_{k \rightarrow \infty} z_{n_k}(s) = \bar{z}(s).$$

Множество $Z \downarrow_C$ является компактом в пространстве L_p , поскольку подпоследовательность $E1$, состоящая из равномерно ограниченных функций, сходится к функции \bar{z} в каждой точке s , т. е. $E1$ сходится в $L_p[a, b]$, $p > 1$. Пусть $Z = L_p[a, b]$, $p > 1$, а оператор \hat{A} взаимно однозначно отображает $Z \downarrow_C$ на $AZ \downarrow_C$, т. е. $z_1, z_2 \in Z \downarrow_C$, а из равенства $\|\hat{A}z_1 - \hat{A}z_2\| = 0$ следует, что $z_1 = z_2$. Зададим множество $Z \downarrow_C(\eta)$ ¹ таких элементов z из $Z \downarrow_C$, что

$$\|\hat{A}_h z - u_\delta\| \leq \psi(h, \|z\|) + \delta.$$

В этом случае любой произвольный элемент z_η из множества $Z \downarrow_C(\eta)$ будет являться приближенным решением задачи (2.1). Более того, при $\eta \rightarrow 0$ в L_p будет наблюдаться сходимость $z_\eta \rightarrow \bar{z}$.

Теорема 5 [8, с. 70–71]. Пусть $\bar{z}(s) \in C[a, b]$, z_n – произвольный элемент из $Z \downarrow_C(\eta_n)$. Пусть $[\gamma, \sigma]$ – произвольный фиксированный отрезок из (a, b) . Тогда при $n \rightarrow \infty$ последовательность $z_n(s)$ сходится к $\bar{z}(s)$ равномерно на $[\gamma, \sigma]$.

Из приведенной формулировки теоремы 5 следует, что точное решение $\bar{z}(s)$ обязательно должно быть непрерывной функцией, а приближенное решение в общем случае не обязано быть непрерывным, т. е. функция $z_\eta(s)$ может быть разрывной монотонной функцией. Тем не менее на любом отрезке $[\gamma, \sigma] \subset (a, b)$ при $\eta \rightarrow 0$ будет наблюдаться равномерная сходимость приближенного решения к точному, т. е. $z_\eta(s) \rightarrow \bar{z}(s)$.

В случае, когда точное решение задачи (2.1) есть кусочно-непрерывная функция, теорема 5 остается справедливой при уточнении, что последовательность приближений будет сходиться к $\bar{z}(s)$ равномерно на каждом замкнутом отрезке, не содержащем точек разрыва функции $\bar{z}(s)$ и точек $s = a$, $s = b$.

Итак, если существует априорная информация, что точное решение \bar{z} некорректно поставленной задачи (2.1) является монотонной ограниченной функцией, то за приближенное решение задачи всегда можно принять произвольный элемент $\bar{z}(s)$ из множества $Z \downarrow_C(\eta)$. При $\eta \rightarrow 0$ в пространстве L_p для кусочно-непрерывной функции $\bar{z}(s)$ на каждом замкнутом отрезке, не содержащем точек разрыва функции $\bar{z}(s)$ и точек $s = a$, $s = b$, имеет место равномерная сходимость последовательности приближенных решений к точному решению, т. е. $z_\eta \rightarrow \bar{z}$.

¹Множество $Z \downarrow_C(\eta)$ определяется заданием элементов A_h, u_δ, δ, h .

2. Множество выпуклых функций. Другой вид априорной информации о качественной характеристике точного решения некорректно поставленной обратной задачи (2.1) состоит в предположении, что точное решение $\bar{z}(s)$ является *выпуклой* или *монотонной выпуклой* функцией.

Для новых множеств будем использовать следующие условные обозначения: \tilde{Z}_C – множество выпуклых (для определенности вверх) функций, ограниченных сверху и снизу соответственно константами C и 0 ; $\tilde{Z} \downarrow_C$ – множество выпуклых (вверх) невозрастающих ограниченных константой C неотрицательных функций: $\tilde{Z} \downarrow_C = Z \downarrow_C \cap \tilde{Z}_C$.

Для случая $\eta \neq 0$ обозначим эти множества как $\tilde{Z}_C(\eta)$, $\tilde{Z} \downarrow_C(\eta)$. Все функции $z_\eta \in \tilde{Z}_C(\eta)$, $\tilde{Z} \downarrow_C(\eta)$ должны удовлетворять условию

$$\|\hat{\mathbf{A}}_h z_\eta - u_\delta\| \leq \psi(h, \|z_\eta\|) + \delta.$$

В этом условии оператор $\hat{\mathbf{A}}_h$ взаимно однозначно отображает $\tilde{Z} \downarrow_C$ (\tilde{Z}_C) на $A\tilde{Z} \downarrow_C$ ($A\tilde{Z}_C$), точное решение $\bar{z}(s) \in \tilde{Z} \downarrow_C$ (либо \tilde{Z}_C), функция ошибки $\eta_n \rightarrow 0$ при $n \rightarrow \infty$. Тогда выполняется следующая теорема.

Теорема 6 [8, с. 73]. Пусть z_n – произвольный элемент из $\tilde{Z} \downarrow_C(\eta_n)$ (либо $\tilde{Z}_C(\eta_n)$), а $[\gamma, \sigma]$ – некоторый отрезок из интервала (a, b) . Тогда последовательность $z_n(s) \rightarrow \bar{z}(s)$ при $n \rightarrow \infty$ равномерно на $[\gamma, \sigma]$.

Из условия $\bar{z}(s) \in \tilde{Z}_C$ следует непрерывность функции $\bar{z}(s)$ в каждой внутренней точке $s \in (a, b)$.

В случае $\bar{z}(s) \in \tilde{Z} \downarrow_C$ теорема 6 может быть усилена. Обозначим через z_n – любой элемент из множества $\tilde{Z} \downarrow_C(\eta_n)$. Тогда на любом отрезке $[a, b - \varepsilon]$, $\varepsilon > 0$ при $n \rightarrow \infty$ имеется равномерная сходимость последовательности $z_n(s) \rightarrow \bar{z}(s)$.

При определенных условиях в обоих случаях имеет место не только равномерная сходимость приближений, но и их производных.

Итак, при $\bar{z}(s) \in \tilde{Z} \downarrow_C(\eta)$ (или $\tilde{Z}_C(\eta)$) за приближенное решение некорректно поставленной задачи (2.1) можно принять любой элемент из множества $\tilde{Z} \downarrow_C$ (или \tilde{Z}_C).

Технология поиска приближенного решения на множествах $Z \downarrow_C$, $\tilde{Z} \downarrow_C$ или \tilde{Z}_C основана на использовании какого-либо процесса минимизации функционала невязки $\Phi(z) = \|\hat{\mathbf{A}}_h z - u_\delta\|^2$. В случае линейных операторов $\hat{\mathbf{A}}$, $\hat{\mathbf{A}}_h$ задача минимизации функционала $\Phi(z)$ на

множествах $Z \downarrow_C$, $\tilde{Z} \downarrow_C$, \tilde{Z}_C — задача выпуклого программирования, при этом достаточно лишь найти такой элемент z_η , принадлежащий одному из указанных множеств, для которого $\Phi(z_\eta) \leq (h C_0 + \delta)^2$, где $C_0 = C\sqrt{b-a}$.

Из теоремы Хелли следует вывод о равномерной ограниченности множества допустимых функций, который является существенным при доказательстве их свойств. Ситуация, когда известна константа, ограничивающая множество допустимых функций снизу (например, $C_2 = 0$, искомые функции неотрицательны), и неизвестна константа C , является типичной для обратных задач математической физики. Исследуем этот случай и откажемся от равномерной ограниченности сверху для каждого из множеств допустимых решений, т. е. для каждого из множеств ограниченных снизу неотрицательных функций, которые условно обозначим как $Z \downarrow$ — монотонные (невозрастающие), \tilde{Z} — выпуклые (для определенности вверх), $\tilde{Z} \downarrow$ — выпуклые (вверх). Пусть линейный и ограниченный оператор \hat{A} известен точно, т. е. $h = 0^1$, тогда определим множество функций $Z \downarrow(\delta)$, $\tilde{Z}(\delta)$ и $\tilde{Z} \downarrow(\delta)$ соответственно из $Z \downarrow$, \tilde{Z} и $\tilde{Z} \downarrow$ таких, что $\|\hat{A}z - u_\delta\| \leq \delta$; δ_n — произвольная последовательность такая, что $\delta_n \rightarrow 0$ и $0 \neq \delta_n$ при $n \rightarrow \infty$.

Можно показать, что все функции z_n в совокупности ограничены, т. е. для любого произвольного элемента $z_n \in \tilde{Z}(\delta_n)$ существует такая константа $C > 0$, что для любых n и всех $s \in [a, b]$ выполнено неравенство $z_n(s) \leq C$. Отсюда следует, что если точным решением задачи выбрана функция, которая на отрезке $[a, b]$ является выпуклой или монотонно выпуклой, то при поиске приближенного решения можно отказаться от ограниченности сверху для множеств $\tilde{Z}_C(\eta)$, $\tilde{Z} \downarrow_C(\eta)$, заменив их соответственно на $\tilde{Z}(\delta)$, $\tilde{Z} \downarrow(\delta)$.

Теорема 7 [8, с. 80]. Пусть z_n — произвольный элемент $\tilde{Z}(\delta_n)$. Тогда существует такая константа $C > 0$, что для всех $z_n(s) \leq C$ при всех $s \in [a, b]$.

Для того чтобы построить приближенное решение некорректно поставленной задачи в случае, когда точное решение принадлежит множеству *выпуклых* или *монотонных и выпуклых* функций, нет необходимости знать константу, ограничивающую точное решение задачи сверху. Далее этот важный факт будем широко нами использовать. Отметим также, что если точное решение является монотонной на $[a, b]$ функцией, то аналогичное утверждение в общем случае не является верным.

¹Считаем, что $Z = L_p[a, b]$, $p > 1$.

3. Множество функций ограниченной вариации. Синонимы этого термина – *полная вариация* или *полное изменение*. Будем использовать термин *вариация функции* ($V_a^b(\varphi) \in [0, +\infty[$), который определим следующим образом.

Вариацией функции $\varphi : [a, b] \rightarrow R^n$ на $[a, b]$ называется величина

$$V_a^b(\varphi) = \sup \left[\sum_{i=1}^n |\varphi(t_i) - \varphi(t_{i-1})| : a = t_0 < \dots < t_n = b, n \in N \right].$$

Свойство вариации функции: если $a \leq x \leq y \leq b$, то выполняется

$$V_a^y(\varphi) = V_a^x(\varphi) + V_x^y(\varphi).$$

Функция $\varphi : [a, b] \rightarrow K$ называется функцией ограниченной вариации, если вариация функции ограничена, т. е. выполняется $V_a^b(\varphi) < \infty$.

Рассмотрим некорректно поставленную задачу

$$\hat{\mathbf{A}}z = u, \quad z \in Z = L_p, \quad p > 1, \quad u \in U, \quad (2.2)$$

где $\hat{\mathbf{A}}$ – непрерывный оператор, действующий из Z в U , $\bar{z}(s)$ – функция ограниченной вариации, которая непрерывна на отрезке $[a, b]$ и является точным решением задачи (2.2). В реальном случае вместо оператора $\hat{\mathbf{A}}$ и точной правой части \bar{u} уравнения нам обычно известны только их приближенные значения: непрерывный оператор $\hat{\mathbf{A}}_h$ и элемент u_δ .

Для решения задачи необходимо построить алгоритм, равномерно приближающий $\bar{z}(s)$ на множестве $V \subset L_p[a, b]$ функций ограниченной вариации. Среди функций, приближенно удовлетворяющих уравнению (2.2), будем искать наиболее *гладкую* функцию $z \in V$, а вариацию $V_a^b(z)$ примем за меру *гладкости*. Такая формулировка типична для обратных задач спектроскопии, спектрометрии, ядерной физики, астрофизики. Для удобства обсуждения имеет смысл закрепить значения функций на одном из концов отрезка $[a, b]$, т. к. из равенства $V_a^b(z_1 - z_2) = 0$ следует, что функции $z_1(s)$ и $z_2(s)$ могут отличаться только на константу. Для определенности будем считать, что нам известно значение точного решения $\bar{z}(a) = 0$ в точке $s = a$. Не уменьшая общности рассуждений положим $\bar{z}(a) = 0$.

Определим множество $V(\eta)$ элементов $z \in V$, для которых выполняются два условия: $z(a) = 0$ и $\|\hat{\mathbf{A}}_h z - u_\delta\| < \psi(h, \|z\|) + \delta$. Будем считать, что оператор $\hat{\mathbf{A}}$ взаимно однозначно отображает V на $AV \subset U$, т. е. из равенства $\hat{\mathbf{A}}z_1 = \hat{\mathbf{A}}z_2$ (пространство U) следует, что $z_1 = z_2$ (пространство L_p).

Лемма 1 [8, с.81]. Пусть $z_n \rightarrow z_0$ при $n \rightarrow \infty$ в каждой точке $s \in [a, b]$, причем $z_n \in V$ и все $V_a^b(z_n)$ ограничены в совокупности. Тогда $z_0 \in V$, причем $V_a^b(z_0) \leq \lim_{n \rightarrow \infty} V_a^b(z_n)$.

Лемма 2 [8, с.81]. Пусть $\bar{z}(s) \in V$, причем $\bar{z}(s)$ непрерывна в $[a, b]$. Пусть \bar{z} и $\bar{\bar{z}}$ совпадают на $[a, b]$ почти всюду. Тогда выполняется $V_a^b(\bar{z}) \leq V_a^b(\bar{\bar{z}})$. Если же $\bar{z}(s)$ не совпадают с $\bar{\bar{z}}(s)$ тождественно на $[a, b]$, то $V_a^b(\bar{z}) < V_a^b(\bar{\bar{z}})$.

Теорема 8 [8, с.81–82]. Функционал $V_a^b(z)$ достигает на $V(\eta)$ своей точной нижней грани.

Теорему 8 можно доказать (см. [8]), предполагая, что все функции $z(s) \in V(\eta)$ имеют закрепленный конец $z(a) = 0$. В случае, когда оператор $\hat{\mathbf{A}}$ из L_p в пространстве U , $p > 1$ линейен, непрерывен и известен точно ($h = 0$), теорему 8 можно также доказать не привлекая предположение, что все функции $z(s) \in V(\eta)$ имеют закрепленный конец $z(a) = 0$.

Теорема 9 [8, с.82–84]. Пусть $\bar{z}(s) \in C[a, b] \cap V$. Тогда $z_\eta \rightarrow \bar{z}$ равномерно на отрезке $[a, b]$ при $\eta \rightarrow 0$.

При наличии априорной информации, что точное решение некорректно поставленной задачи (2.2) есть непрерывная функция ограниченной вариации, за приближенное решение можно взять любой элемент z_η , который реализует минимум функционала $V_a^b(z)$ на множестве $V(\eta)$. В этом случае можно гарантировать равномерную сходимость последовательности приближенных решений к точному решению задачи (2.2).

В случае, когда линейный непрерывный оператор $\hat{\mathbf{A}} \in L_p$ известен точно ($h = 0$), для доказательства теоремы 9 не требуется даже предположения, что все функции $z(s) \in V(\eta)$ имеют закрепленный конец.

2.3. Конечно-разностные аналоги компактных множеств в L_2

Рассмотрим некорректно поставленную задачу (2.3), для которой существует априорная информация о том, что решение этой задачи следует искать на компактных множествах $Z \downarrow_C$, \bar{Z}_C или $\bar{\bar{Z}} \downarrow_C$.

$$\hat{\mathbf{A}}z = u, \quad z \in Z, \quad u \in U. \quad (2.3)$$

При построении алгоритмов приближенного решения задачи (2.3) в дальнейшем будут использованы некоторые специфические свойства конечно-разностной аппроксимации упомянутых компактных множеств, а при численной реализации этих алгоритмов будут использованы равномерные сетки, определенные на отрезке $[a, b]$.

Конечно-разностная аппроксимация некорректно поставленной задачи (2.3) предполагает, что функционал невязки

$$\Phi(z) = \|\hat{\mathbf{A}}z - u_\delta\|^2, \quad (2.4)$$

определенный на одном из компактных множеств $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$, заменяется его конечно-разностной аппроксимацией, т.е. функцией $\Phi(z_i)$, которая определена на R^n . Компакты $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$ при этом переходят соответственно в следующие множества векторов из пространства R^n :

$$M \downarrow_C = \left\{ \begin{array}{l} z_{i+1} - z_i \leq 0, \quad i = 1, 2, \dots, n-1 \\ 0 \leq z_i \leq C, \quad i = 1, 2, \dots, n \end{array} \right\}, \quad (2.5)$$

здесь $z_i = z(s_i) \in R^n$ – значение функции $z(s)$ в узлах сетки.

$$\tilde{M}_C = \left\{ \begin{array}{l} z_{i-1} - 2z_i + z_{i+1} \leq 0, \quad i = 2, 3, \dots, n-1 \\ 0 \leq z_i \leq C, \quad i = 1, 2, \dots, n \end{array} \right\}; \quad (2.6)$$

$$\tilde{\tilde{M}} \downarrow_C = \left\{ \begin{array}{l} z_{i-1} - 2z_i + z_{i+1} \leq 0, \quad i = 2, 3, \dots, n-1 \\ z_{i+1} - z_i \leq 0, \quad i = 1, 2, \dots, n-1 \\ 0 \leq z_i \leq C, \quad i = 1, 2, \dots, n \end{array} \right\}. \quad (2.7)$$

В пространстве R^n каждое из множеств $M \downarrow_C$, \tilde{M}_C или $\tilde{\tilde{M}} \downarrow_C$ представляет собой замкнутый *выпуклый* ограниченный многогранник. В дальнейшем некоторые свойства этих многогранников, заданных в пространстве R^n , понадобятся для отыскания координат вершин этих многогранников.

В векторном пространстве над полем вещественных чисел зададим конечное число точек x_1, x_2, \dots, x_n (точки – это векторы, скаляры, точки *аффинного* пространства и т.п.). Выпуклая комбинация этих точек по определению будет иметь вид $\alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$, где коэффициенты α_i – вещественные числа: $\alpha_i \geq 0$; $\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$.

Если ни один из векторов $z_j \in R^n$, ($j = 1, 2, \dots, m$) не является выпуклой комбинацией остальных, то такие векторы называются *выпукло независимыми* векторами.

Выпуклый многогранник в R^n – выпуклая оболочка конечного числа векторов $z_j \in R^n$, ($j = 1, 2, \dots, l$). Точки $z_j \in R^n$ – вершины многогранника

$$M = C \{z_j\} \Big|_{j=1}^l = \left\{ x : x = \sum_{j=1}^l a_j z_j, \quad a_j \geq 0, \quad \sum_{j=1}^l a_j = 1 \right\},$$

если существует такой вектор $b \in R^n$, что для любого вектора $x \in M$ справедливо неравенство $(b, x - z) \geq 0$ и равенство достигается лишь при $x = z$. Иллюстрация определения приведена на рис. 2.3.

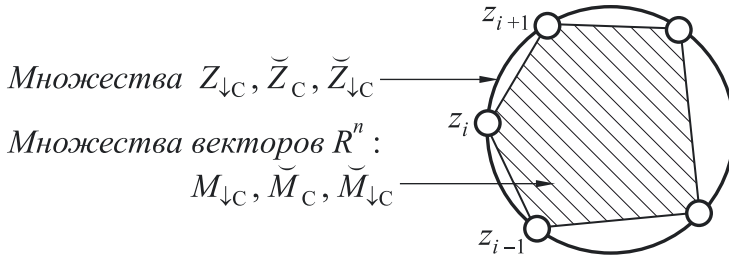


Рис. 2.3. Графическая иллюстрация определения выпуклого многогранника

Можно доказать следующие элементарные утверждения.

Лемма 3 [8, с. 74]. Дан выпуклый многогранник $M = C \{z_i\} \Big|_{i=0}^m$, пусть t – вершина M . Тогда существует такое p : $0 \leq p \leq m$, что $z_p = t$.

Лемма 4 [8, с. 74]. Дан выпуклый многогранник $M = C \{z_i\} \Big|_{i=0}^m$. Тогда для того, чтобы элемент $z_0 \in M$ являлся вершиной M , необходимо и достаточно, чтобы этот элемент z_0 не являлся выпуклой комбинацией векторов z_i ($i = 0, 1, \dots, m$).

Лемма 5 [8, с. 74]. Всякий выпуклый многогранник есть выпуклая оболочка своих вершин.

При необходимости выяснить, дает ли набор векторов z_i ($i = 0, 1, \dots, m$) все вершины заданного выпуклого многогранника $M \subset R^n$, требуется проверить выполнение следующих двух условий:

а) любой вектор $z \in M$ можно представить в виде выпуклой комбинации z_i ($i = 0, 1, \dots, m$);

б) векторы z_i ($i = 0, 1, \dots, m$) выпукло независимы.

Теорема 10 [8, с. 74–75]. Множество $M \downarrow_C$ представляет собой выпуклый многогранник в R^n , вершины которого $T^{(j)}$ ($j = 0, 1, \dots, n$) можно записать в явном виде:

$$T^{(0)} = 0; \quad T_i^{(j)} = \begin{cases} C, & i \leq j, \\ 0, & i > j, \end{cases} \quad j = 1, 2, \dots, n. \quad (2.8)$$

Теорема 11 [8, с. 76–77]. Множество $\tilde{M} \downarrow_C$ представляет собой выпуклый многогранник в R^n , вершины которого $T^{(j)}$ ($j = 0, 1, \dots, n$) можно записать в явном виде:

$$T^{(0)} = 0; \quad T_i^{(j)} = \begin{cases} C, & i \leq j, \\ \frac{n-i}{n-j} C, & i > j, \end{cases} \quad j = 1, 2, \dots, n. \quad (2.9)$$

Теорема 12 [8, с. 77]. Множество $\tilde{M}_C \subset R^n$ представляет собой выпуклый многогранник в R^n , вершины которого $T^{(i,j)}$ $0 \leq i \leq j \leq n$ можно записать в явном виде:

$$T^{(0,0)} = 0; \quad T_k^{(i,j)} = \begin{cases} \frac{k-1}{i-1} C, & k < i, \\ C, & i \leq k \leq j, \\ \frac{n-k}{n-j} C, & k > j, \end{cases} \quad 1 \leq i \leq j \leq n.$$

Из теорем 10–12 видно, что для каждого выпуклого ограниченного многогранника $M \downarrow_C$, \tilde{M}_C или $\tilde{M} \downarrow_C$, определенного в пространстве R^n , можно в явном виде записать координаты вершин. При этом многогранники $M \downarrow_C$ и $\tilde{M} \downarrow_C$ имеют по $n + 1$ вершине, а многогранник \tilde{M}_C имеет $n(n + 1)/2 + 1$ вершину. Невозможно наглядно представить эти многогранники в общем случае n -мерного пространства, однако в частном случае трехмерного пространства ($n = 3$) геометрические образы компактов $M \downarrow_C$, \tilde{M}_C и $\tilde{M} \downarrow_C$ можно наглядно продемонстрировать (см. рис. 2.4).

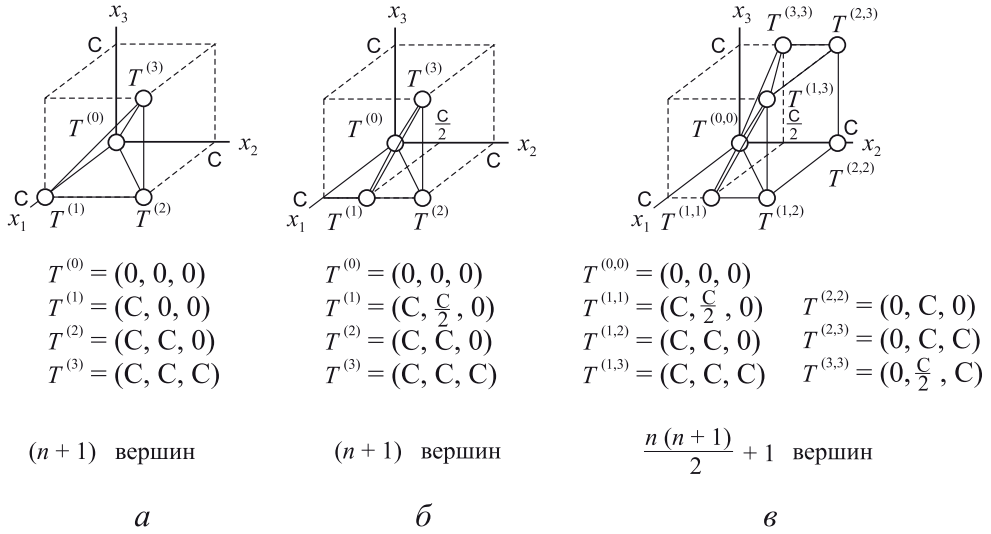


Рис. 2.4. Геометрические образы компактов $M \downarrow_C$ (а), \tilde{M}_C (б) и \tilde{M}'_{2C} (в) в частном случае $n = 3$. Для каждого компакта указаны координаты вершин

Очень часто для минимизации количества вершин многогранник \tilde{M}_C аппроксимируют многогранником \tilde{M}'_{2C} , который содержит только $n + 1$ вершину. Аппроксимирующий многогранник \tilde{M}'_{2C} в этом случае обладает следующими важными свойствами.

Теорема 13 [8, с. 77–78]. *Множество \tilde{M}_C содержится в выпуклом многограннике \tilde{M}'_{2C} , вершины которого $T^{(j)}$, ($j = 0, 1, \dots, n$) записываются в виде:*

$$T^{(0)} = 0; \quad T_i^{(j)} = \begin{cases} 2C \frac{i-1}{j-1}, & i < j, \\ 2C, & i = j, \\ 2C \frac{n-i}{n-j}, & i > j, \quad j = 1, 2, \dots, n. \end{cases}$$

Из теоремы 13 очевидно, что если точное решение задачи принадлежит компактному множеству \tilde{M}_C , то приближенное решение можно искать на множестве \tilde{M}'_{2C} . Обоснование возможности такой аппроксимации базируется на простых соображениях. Действительно, качественная информация о выпуклости вверх функций сохраняется, поскольку $\tilde{M}'_{2C} \subset \tilde{M}_{2C}$, а множество функций из \tilde{M}'_{2C} ограничено сверху константой $2C$, поэтому все теоремы о сходимости последова-

тельности приближений к точному решению задачи при замене \tilde{M}_C на \tilde{M}'_{2C} остаются в силе.

2.4. Основы построения алгоритма регуляризации

1. Постановка задачи. Рассмотрим процедуру решения интегрального уравнения Фредгольма 1-го рода с гладким ядром

$$\hat{\mathbf{A}}z \equiv \int_a^b K(t, s) z(s) ds = u(t), \quad c \leq t \leq d. \quad (2.10)$$

Здесь использованы следующие условные обозначения: $\hat{\mathbf{A}}$ – линейный интегральный оператор; $K(t, s)$ – ядро интегрального уравнения, известная функция; $u \in U$ (U – гильбертово пространство) – известная функция правой части уравнения; $z(s) \in D \subset Z$ – искомая функция, принадлежащая одному из специальных множеств ($Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$). Ранее было показано, что в случае, когда \tilde{z} принадлежит $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$, за приближенное решение задачи (2.13) можно принять произвольный элемент z_δ соответственно из $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$, для которого $\|\hat{\mathbf{A}}z_\delta - u_\delta\| \leq \delta$.

2. Метод условного градиента. Это сравнительно простой и быстрый метод решения задач выпуклого программирования. Пусть дано

$$\begin{cases} f(z) \rightarrow \inf, \\ z \in D, \end{cases} \quad (2.11)$$

где $f: R^n \rightarrow R^n$ – выпуклая функция, которую оптимизируют на замкнутом выпуклом множестве $Z \subset R^n$, начиная процесс с точки $z^0 \in Z$. Градиентный метод необходимо модифицировать с учетом ограничения ($z \in D$). Для произвольной $z^k \in Z$ имеем разложение f в окрестности точки z^k

$$f(z) = f(z^k) + \underbrace{(f'(z^k), z - z^k)}_{f_k(z)} + o(z - z^k).$$

Линейную часть приращения $f_k(z)$ с точностью до o -малого остатка можно определить как $f_k(z) = f(z) - f(z^k)$. Вместо задачи (2.11)

рассмотрим задачу на нахождение минимума линейного члена $f_k(z)$

$$\begin{cases} f_k(z) \rightarrow \inf, \\ z \in D. \end{cases} \quad (2.12)$$

Из линейности функции $f_k(z)$ следует, что решение задачи (2.12), которое обозначим через \bar{z}^k , должно лежать на границе множества $D \subseteq Z$.

Если минимум $f_k(z)$ достигается в точке $\bar{z}^k = z^k$, то можно видеть, что в любых направлениях, исходящих из точки z^k и не выходящих при достаточно малых шагах из допустимого множества D , функция f возрастает. Это означает, что z^k является решением задачи (2.11).

Напротив, если $\bar{z}^k \neq z^k$, то рассмотрим направление $p_k = \bar{z}^k - z^k$. Линейная часть приращения $f_k(z)$ будет убывать вдоль вектора p_k . Иными словами, найдется малое положительное число λ такое, что $f(z^k + \lambda p_k) < f(z^k)$. Более того, из-за выпуклости множества $D \subseteq Z$ получим, что при $\lambda \in [0, 1]$

$$z^k + \lambda p_k = z^k + \lambda(\bar{z}^k - z^k) = \lambda \bar{z}^k + (1 - \lambda) z^k \in D.$$

Используя эту формулу, можно найти такое значение искомого параметра λ_k , что

$$f(z^k + \lambda_k p_k) - f(z^k) < 0, \quad z^k + \lambda_k p_k \in D.$$

Получили распространение два способа определения λ_k на практике:

- 1) $\lambda_k = \min\{1, \lambda^*\}$, где $\lambda^* = \arg \min f(z^k + \lambda p_k)$ при $\lambda > 0$;
- 2) (а) принимают $\bar{\lambda} = 1$, выбирают $\delta \in (0, 1)$;
- (б) проверяют $f(z^k + \lambda p_k) - f(z^k) < 0$;
- (в) при выполнении условия (б) полагают $\lambda_k = \bar{\lambda}$, иначе дробят $\bar{\lambda}$ с помощью δ (т. е. принимают $\bar{\lambda} = \delta \bar{\lambda}$) и возвращаются к шагу (б).

Во втором случае итерационный процесс продолжают до тех пор, пока не выполнится критерий остановки, в качестве которого можно взять, например, $\|f'(z^k)\| < \varepsilon$, где $\varepsilon > 0$ – точность решения.

3. Решение некорректной задачи. Рассмотрим некоторые особенности решения некорректно поставленной задачи

$$\hat{A}z = u, \quad z \in D \subset Z = L_2, \quad u \in U, \quad (2.13)$$

где \hat{A} – линейный непрерывный оператор, действующий из Z в U , точно ($h = 0$) известен; U – гильбертово пространство. В качестве

множества D можно рассматривать одно из компактных множеств $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$. В случае, когда \bar{z} принадлежит $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$, за приближенное решение задачи (2.13) можно принять любой произвольный элемент z_δ соответственно из $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$ для которого $\|\hat{\mathbf{A}} z_\delta - u_\delta\| \leq \delta$.

Каждое из выпуклых множеств $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$ в пространстве Z будет ограниченным и замкнутым множеством. Рассмотрим функционал

$$\Phi(z) = \|\hat{\mathbf{A}} z - u_\delta\|^2, \quad (2.14)$$

определенный для всех z , принадлежащих соответственно $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$. Из линейности оператора $\hat{\mathbf{A}}$ следует, что $\Phi(z)$ будет квадратичной функцией переменной z . Отметим также, что функционал $\Phi(z)$ является выпуклым и дифференцируемым, а его производная *Фреше* равна

$$\Phi'(z) = 2 \left(\hat{\mathbf{A}}^* \hat{\mathbf{A}} z - \hat{\mathbf{A}}^* u_\delta \right) = 2 \hat{\mathbf{A}}^* (\hat{\mathbf{A}} z - u_\delta), \quad (2.15)$$

где $\hat{\mathbf{A}}^*$ – сопряженный оператор, действующий из U в Z , для которого выполняется условие Лагранжа $(\hat{\mathbf{A}} x, y) = (x, \hat{\mathbf{A}}^* y)$. Заметим, что

$$\|\Phi'(z_1) - \Phi'(z_2)\| = 2 \|\hat{\mathbf{A}}^* \hat{\mathbf{A}}(z_1 - z_2)\| \leq 2 \|\hat{\mathbf{A}}\|^2 \|z_1 - z_2\|.$$

Из этого неравенства видно, что производная *Фреше* функционала $\Phi(z)$ удовлетворяет условию *Липшица* с константой $L = 2 \|\hat{\mathbf{A}}\|^2$.

Решить задачу на отыскание приближенного решения в нашем случае можно путем минимизации функционала $\Phi(z)$ соответственно на множествах $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$. Важно, что при этом нет необходимости определять значение минимума $\Phi(z)$ на этих множествах. Для решения достаточно найти элемент z_δ из указанных множеств такой, что $\Phi(z_\delta) \leq \delta^2$, т.е. для отыскания приближенного решения (2.13) на множествах $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$ необходимо только построить последовательность, минимизирующую выпуклый дифференцируемый функционал на замкнутом выпуклом ограниченном множестве в гильбертовом пространстве.

В нашем случае производная *Фреше* этого функционала удовлетворяет условию *Липшица*, поэтому для решения задачи минимизации функционала можно использовать, например, *метод условного градиента*.

4. Минимизация функционала методом условного градиента. Задачу минимизации рассмотрим на примере функционала (2.4)

$$\Phi(z) = \|\hat{\mathbf{A}} z - u_\delta\|^2.$$

Для минимизирующей последовательности $z = z^{(k)}$ (номер итерации $k = 0, 1 \dots, n$) в окрестности каждой $z^{(k)}$ функционал $\Phi(z)$ представим в виде

$$\Phi(z) = \Phi(z^{(k)}) + \Phi^{(k)}(z), \quad \text{где} \quad \Phi^{(k)}(z) \equiv \left(\Phi'(z^{(k)}), (z - z^{(k)}) \right).$$

По условию разложения в ряд значения $\Phi(z^{(k)})$ близки к $\Phi(z)$, поэтому будем минимизировать линейный член $\Phi^{(k)}(z)$. Наряду с минимизирующей последовательностью $z^{(k)}$, начинающейся с произвольной допустимой точки $z^{(0)} \in M^1$, в методе условного градиента строится вспомогательная последовательность $\bar{z}^{(k)}$, являющаяся результатом минимизации линейного члена. Если $z^{(k)}$ уже построено, то $\bar{z}^{(k)}$ будет решением задачи

$$\bar{z}^{(k)} = \arg \min_{z \in M} \Phi^{(k)}(z). \quad (2.16)$$

Решение задачи 2.16 существует, поскольку существует точка $\bar{z}^{(k)}$, в которой линейный функционал $\Phi^{(k)}(z)$ достигает своего минимального значения на множестве M . Однако это решение не является единственным. Ранее было показано, что для линейного функционала решение $\bar{z}^{(k)}$ принадлежит границе множества M (рис. 2.5). Если ограниченное замкнутое множество M в пространстве R^n является выпуклым многогранником, то задача (2.16) решается совсем просто. В этом случае задача (2.16) превращается в обычную задачу линейного программирования, которая решается симплекс-методом. Если, кроме того, известны точные координаты вершин многогранника M , то задачу (2.16) можно решить простым перебором всех его вершин.

Для каждой найденной $\bar{z}^{(k)}$ можно определить следующий элемент итерационной последовательности, используя формулу

$$z^{(k+1)} = z^{(k)} + \lambda_k \left(\bar{z}^{(k)} - z^{(k)} \right). \quad (2.17)$$

Минимизация $\Phi(z)$ на отрезке $[z^{(k)}, \bar{z}^{(k)}]$ завершает текущую итерацию:

$$\lambda_k = \arg \min_{\lambda \in [0, 1]} \Phi \left(z^{(k)} + \lambda (\bar{z}^{(k)} - z^{(k)}) \right). \quad (2.18)$$

¹ Допустимая точка—произвольный элемент $z \in M$. Так как все множества $Z \downarrow_C$, \bar{Z}_C или $\bar{Z} \downarrow_C$ содержат 0, то в качестве допустимой точки можно взять $z^{(0)} = 0$.

Из выпуклости множества M следует, что $z^{(k+1)} \in M$, т.е. итерационный процесс, начатый с $z^{(0)} \in M$, не может выйти за пределы множества M .

Если оператор \hat{A} линеен, то $\Phi(z)$ будет квадратичной функцией z и задача (2.18) сведется к элементарному поиску минимума параболы по λ на отрезке $[0, 1]$. При этом построенная последовательность $z^{(k)}$ будет минимизирующей для функционала $\Phi(z)$ на множестве M .

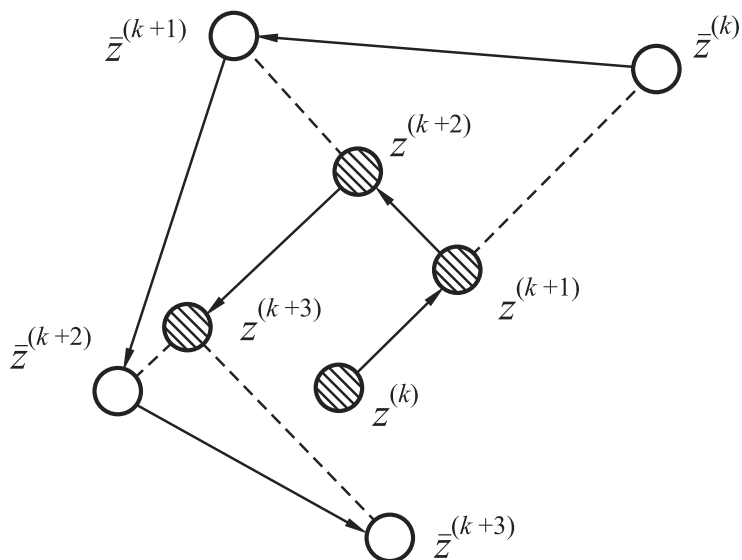


Рис. 2.5. Минимизация функционала методом условного градиента

Поскольку в нашем случае $\Phi(z)$ является квадратичной функцией и задача (2.18) решается элементарно, то для построения эффективного алгоритма приближенного решения некорректно поставленной задачи (2.13) на множествах $Z \downarrow_C$, \tilde{Z}_C или $\tilde{Z} \downarrow_C$ необходимо только научиться эффективно решать задачу (2.16). Конечно-разностная аппроксимация множеств $Z \downarrow_C$, \tilde{Z}_C или $\tilde{Z} \downarrow_C$ состоит в замене их на многогранники $M \downarrow_C$, $\tilde{M} \downarrow_C$, \tilde{M}_C , вершины которых известны. Поэтому задачу (2.16) можно решить простым перебором вершин указанных многогранников. На множестве выпуклых функций многогранник \tilde{M}_C можно аппроксимировать многогранником \tilde{M}_{2C} , который имеет всего $n + 1$ вершину, тогда решение задачи (2.16) значительно упрощается. Итак, во всех случаях задача линейного программирования (2.16) решается на каждой итерации за n шагов простым перебором вершин указанных многогранников.

Пусть теперь оператор $\hat{\mathbf{A}}$ задан приближенно и известно, что точное решение задачи (2.13) принадлежит одному из множеств $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$. Тогда за приближенное решение можно принять произвольный элемент z_η из компактного множества $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$, для которого $\|\hat{\mathbf{A}}_h z_\eta - u_\delta\| \leq \psi(h, \|z_\eta\|) + \delta$, а если операторы $\hat{\mathbf{A}}$, $\hat{\mathbf{A}}_h$ линейны, то $\|\hat{\mathbf{A}}_h z_\eta - u_\delta\| \leq h \|z_\eta\| + \delta$. В последнем неравенстве норму z_η можно заменить ее оценкой сверху постоянной $C_0 = \|z\|$ соответственно на множествах $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$. Константу C_0 можно выбрать, например, $C_0 = C(b-a)^{1/2}$. С учетом этого алгоритмы отыскания приближенного решения некорректно поставленной задачи (2.13) на множествах $Z \downarrow_C$, \tilde{Z}_C или $\tilde{\tilde{Z}} \downarrow_C$ будут выглядеть так же, как и в случае, когда оператор $\hat{\mathbf{A}}$ известен точно.

2.5. Формулировка модельной задачи

В качестве модельной задачи будем рассматривать интегральное уравнение (2.10)

$$\int_a^b K(t, s) z(s) ds = u(t), \quad t \in [c, d] \quad (2.19)$$

с гладким ядром

$$K(t, s) = \frac{1}{1 + 100(t-s)^2} \quad (2.20)$$

и значениями пределов интегрирования $a = 0$, $b = 1$, $c = -1$, $d = 2$.

Для аппроксимации интегрального оператора в уравнении (2.10) запишем матрицу линейного оператора $\hat{\mathbf{A}}$ в виде

$$A_{i,j} = \begin{cases} h_s K(t_i, s_j), & j = 2, 3, \dots, n-1, \\ \frac{h_s}{2} K(t_i, s_j), & j = 1, n, \end{cases} \quad (2.21)$$

где $h_s = (b-a)/(n-1)$ – шаг равномерной сетки $\{s_j\}_{j=1}^n$ на отрезке $[a, b]$, $s_1 = a$, $s_n = b$; $i = 1, 2, \dots, m$.

Если необходимо использовать более точные аппроксимационные формулы, то достаточно изменить лишь подпрограмму `amat`, осуществляющую переход к конечно-разностной задаче и формирующую элементы матрицы $[A]$.

Правую часть $u(t)$ уравнения (2.10) зададим на равномерной сетке $\{t_i\}_{i=1}^m$ с шагом $h_t = (c - d)/(m - 1)$ на отрезке $t \in [c, d]$, $t_1 = c$, $t_m = d$.

Вектор \bar{u} правой части получим умножением матрицы $[A]$ размером $m \times n$, которая аппроксимирует оператор в (2.10), на вектор-столбец \bar{z} значений точного решения на сетке $\{s_j\}_{j=1}^n$ на отрезке $[a, b]$:

$$\bar{u}_i = \sum_{j=1}^n A_{i,j} \bar{z}(s_j). \quad (2.22)$$

При данном способе выбора правой части минимум функционала невязки $\Phi(z) = \|\hat{\mathbf{A}}z - \bar{u}\|^2$ на соответствующем множестве векторов будет равен нулю. Вектор $u_{\delta i}$ получим из правой части (2.10) путем возмущения случайной погрешностью, равномерно распределенной на отрезке $[-\delta_0, \delta_0]$, где δ_0 составляет 1% от максимального значения правой части. Тогда среднеквадратическая погрешность $u_{\delta}(t)$ соответствует значению

$$\delta = \left(\frac{1}{n-1} \sum_{i=1}^n (\bar{u}_i - u_{\delta i})^2 \right)^{1/2}.$$

Параметры модельной задачи выберем таким образом, чтобы погрешность аппроксимации задачи (2.10) системой линейных уравнений

$$\psi(z) = \left(h_t \sum_{i=1}^m \left[\int_a^b K(t_i, s) z(s) ds - \sum_{j=1}^n A_{i,j} z(s_j) \right]^2 \right)^{1/2},$$

где $h_t = (d - c)/(m - 1)$ – шаг равномерной сетки по t на $[c, d]$, $t_1 = c$, $t_m = d$, на точном решении $\psi(\bar{z})$ составила $\sim 10^{-4}$. Тогда полученное значение погрешности аппроксимации будет соответствует относительной погрешности $\psi(\bar{z}) / \|\bar{u}\|_{L_2} = 0.001$. Рассмотрим результаты модельных расчетов.

1. Расчет для уравнения (2.19), выполненный на множестве *монотонно убывающих* функций (рис. 2.6). В качестве точного решения уравнения (2.19) выбрана $\bar{z}(s) = 1 - s^2$, правая часть (2.19) возмущена случайной погрешностью, равномерно распределенной на отрезке $[-\delta_0, \delta_0]$, где δ_0 составляет 1% от максимума значений правой части. Это значение среднеквадратической погрешности будет соответствовать $\delta^2 = 8.31 \times 10^{-7}$. Использовали начальное приближение

$z^{(0)}(s) = 0$ и константа $C = 1.045$. Приближенное решение, найденное после 400 итераций, и точное решение соответствуют значению функционала невязки $\Phi(z) = 3.61 \times 10^{-6}$. Дальнейшее увеличение количества итераций в методе условного градиента мало изменяет величину невязки и приближенное решение.

2. Расчет той же модельной задачи (2.19), выполненный на множестве *монотонно выпуклых* функций (рис. 2.7). Точное решение выбрано в виде $\bar{z}(s) = 1 - s^2$. Уровень возмущения входных данных составил 3% от максимального уровня правой части. Начальное приближение $z^{(0)}(s) = 0$, $C = 0.965$. Полученное при этом приближенное решение соответствует значению функционала невязки 2.15×10^{-5} и получено после 50 итераций метода условного градиента.

3. Расчет для уравнения (2.19), выполненный на множестве *выпуклых* функций (рис. 2.8). Точное решение выбрано в виде функции $\bar{z}(s) = 4s(1 - s)$, правая часть возмущена таким образом, что $\delta^2 = 3.47 \times 10^{-6}$. Начальное приближение $z^{(0)}(s) = 0$, $C = 1.000$. Приближенное решение, полученное после 800 итераций метода условного градиента, соответствует значению функционала невязки 1.29×10^{-5} , что составляет примерно 1% от максимума правой части.

4. Расчет для уравнения (2.19), выполненный на множестве функций *ограниченной вариации* V (рис. 2.9). Для расчета был применен метод условного градиента на множестве функций ограниченной вариации V , численное значение вариации которых не превосходило заданной константы $C = 10$. Начальное приближение выбирали в виде $z^{(0)}(s) = 0$.

Точное решение задачи (2.19) выбрано в виде суммы двух гауссовых функций при значениях констант $D_0 = 0.9550408$; $D_1 = 0.0521309$:

$$\bar{z}(s) = D_0 \left[\exp \left(-\frac{(s - 0.3)^2}{0.03} \right) + \exp \left(-\frac{(s - 0.7)^2}{0.03} \right) \right] - D_1.$$

Правая часть уравнения (2.19) была возмущена равномерно распределенной случайной погрешностью таким образом, что $\delta^2 = 2.2 \times 10^{-6}$. Приближенное решение $z(s)$ было достигнуто после 800 итераций. Значение функционала невязки $\Phi(z) = 9.18 \times 10^{-6}$ соответствует примерно 1.5% от максимального значения правой части.

Все рассмотренные методы были реализованы с использованием программного обеспечения, описанного в п. 2.6, реализующего метод условного градиента, поскольку в случае линейного оператора \hat{A} переход к конечно-разностной аппроксимации соответствует задаче минимизации квадратичного функционала на компактном множестве, которое представляет собой выпуклый многогранник в пространстве R^n .

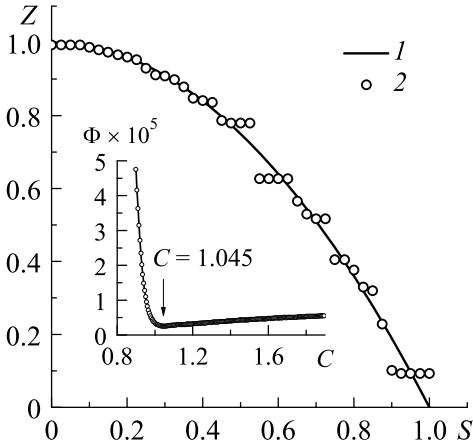


Рис. 2.6. Точное (1) и приближенное (2) решения (400 итераций; невязка $\Phi(z) = 3.61 \times 10^{-6}$; монотонно убывающие функции)

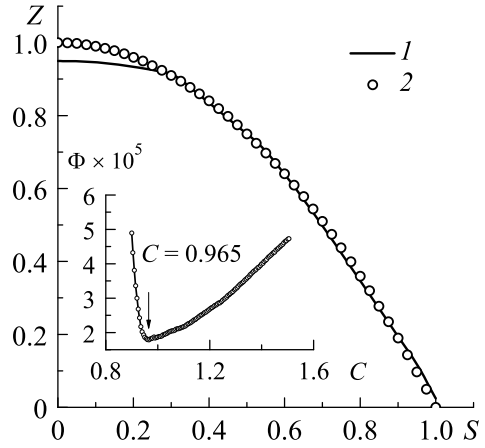


Рис. 2.7. Точное (1) и приближенное (2) решения (50 итераций алгоритма; невязка $\Phi(z) = 2.15 \times 10^{-5}$; монотонно выпуклые функции)

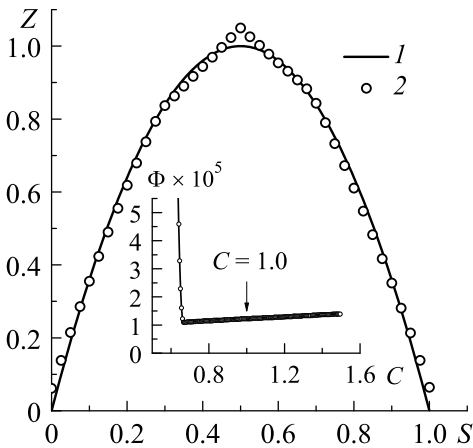


Рис. 2.8. Точное (1) и приближенное (2) решения (800 итераций; невязка $\Phi(z) = 1.29 \times 10^{-5}$; множество выпуклых функций)

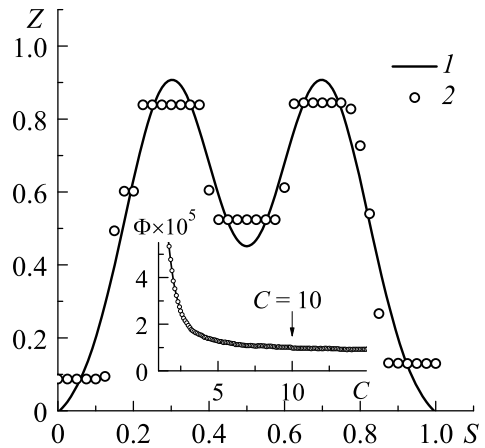


Рис. 2.9. Точное (1) и приближенное (2) решения (800 итераций; невязка $\Phi(z) = 9.18 \times 10^{-6}$; функции ограниченной вариации)

В каждом случае координаты вершин многогранника удастся найти в явном виде. Отметим, что наряду с методом условного градиента для решения некорректно поставленных задач на множествах специальной структуры применяют другие методы минимизации: метод проекций сопряженных градиентов, метод проекции сопряженных градиентов с проецированием на множество векторов с неотрицательными компонентами и т. п.

2.6. Описание программного обеспечения

Программное обеспечение для изучения методов регуляризации на специальных множествах реализовано в вычислительной среде *Мат-лаб* и представлено двумя основными (*головными*) программами и несколькими вспомогательными программами.

Головные программы. 1. *Compact* – осуществляет минимизацию методом условного градиента на множествах специальной структуры. В качестве одного из специальных множеств для решения задачи (2.19) здесь можно выбрать множество функций: монотонных, монотонно выпуклых, выпуклых или ограниченной вариации.

2. *Zadanie2* – это тестовый пример для демонстрации интерфейса доступа к программе *Compact*.

Для осуществления конечно-разностной аппроксимации интегрального уравнения (2.10) использованы обозначения векторов и матриц:

A – матрица $[m \times n]$ оператора;

Z – вектор-столбец $[n \times 1]$ решения уравнения;

U – вектор-столбец $[m \times 1]$ правой части уравнения.

2.6.1. Минимизация методом условного градиента

Рассмотрим квадратичный функционал следующего вида

$$\varphi(z) = \sum_{i=1}^m \left(\sum_{j=1}^n A_{ij} z_j - u_i \right)^2, \quad (2.23)$$

определенный на одном из специальных множеств $Z \downarrow_C$, \check{Z}_C или $\check{\check{Z}} \downarrow_C$. Конечно-разностную аппроксимацию каждого из таких множеств будем обозначать $M \downarrow_C$, \check{M}_C или $\check{\check{M}} \downarrow_C$, соответственно. В некоторых случаях будем указывать эти множества без конкретизации их типа – M . Это будет означать, что сказанное относится в равной мере

к любому из них. Минимизацию квадратичного функционала будем осуществлять методом условного градиента, алгоритм которого реализован в модуле **Compact**. Последовательность выполнения этапов алгоритма следующая.

1. Задать произвольную допустимую точку $z = z^{(0)} \in M$ с которой начнем строить минимизирующую последовательность $z^{(k)}$. Допустимая точка – это произвольный элемент $z \in M$. Так как все множества $Z \downarrow_C$, \tilde{Z}_C или $\tilde{Z} \downarrow_C$ содержат нулевой элемент, то в качестве допустимой точки можно взять $z^{(0)} = 0$.

2. Вычислить правую часть U , соответствующую выбранному элементу $z = z^{(0)}$:

$$\hat{A}z = u; \quad u_i = \sum_j A_{ij} z_j^{(0)}; \quad U = A \times Z.$$

3. Вычислить значение функционала невязки на начальном приближении $z = z^{(0)}$

$$\varphi(z) = \sum_{i=1}^m \left(\sum_{j=1}^n A_{ij} z_j - u_i \right)^2_{R^m}.$$

4. Вычислить производную Фреше (вектор-градиент) функционала невязки $\varphi = \|\hat{A}z - u_0\|_{R^m}^2$ на начальном приближении $z = z^{(0)}$

$$g = 2 \hat{A}^* (\hat{A}z - u_0).$$

Если обозначить $u = \hat{A}z$ – значение оператора на точке z , то

$$g_j = 2 \sum_{i=1}^m A_{ji} (u - u_0); \quad G = 2 A^T (U - U_0);$$

Здесь использовано обозначение $[A_{ji}]$ для матрицы сопряженного оператора \hat{A}^* : $A_{ji} \equiv A_{ij}^T$. Иными словами, $[A^*] = [A]^T$.

5. Вычислить квадрат нормы градиента $\|g\|^2$, а затем проверить условия окончания работы алгоритма и, если они не выполнены, то перейти к осуществлению итерационного процесса:

а) вычислить вершину H многогранника, определяемого значением параметра ИС, в которой достигается минимальное значение линейного функционала

$$(g, t) = \psi(t).$$

Детали вычисления вершины многогранника приведены в п. 2.6.2;

б) вычислить вектор, направленный из найденной вершины в текущее приближенное решение, т. е. направление, противоположное направлению спуска;

в) минимизировать функционал невязки $\varphi(z)$ на отрезке $z - \lambda h$, где $\lambda \in [0, 1]$; результатом является величина шага $\bar{\lambda}$;

г) изменить приближенное решение в соответствии с найденным шагом $\bar{\lambda}$;

д) снова вычислить значение оператора на новом приближении, градиент в новой точке, значение невязки и квадрат нормы градиента;

е) произвести проверку условий окончания работы программы: если условия окончания не выполнены, то проверяется, уменьшилось ли значение функционала невязки $\varphi(z)$, если не уменьшилось, то алгоритм завершает свою работу с кодом 1; в противном случае запоминают текущее значение невязки и продолжают итерационный процесс.

2.6.2. Выбор оптимальной вершины многогранника

Вспомогательный модуль выбора оптимальной вершины многогранника вызывается программой **Compact** и предназначен для выбора оптимальной вершины t_0 многогранника M_C , т. е. вершины, в которой достигается минимальное значение функционала

$$\psi(t_0) = (g, t_0) = \min_{t \in M_C} \psi(t) = \min_{t \in M_C} (g, t),$$

где g – заданный вектор, который при решении задачи соответствует вектор-градиенту; t – радиус-вектор текущей вершины.

Поиск оптимальной вершины t_0 среди вершин $T^{(j)}$ ($j = 0, 1, \dots, n$), заданных их радиус-векторами, осуществляется простым перебором. Координаты вершин $T^{(j)}$ легко записываются в явном виде. При реализации перебора нет необходимости каждый раз явно вычислять координаты вершин $T^{(j)}$ ($j = 0, 1, \dots, n$) и величину скалярного произведения $(g, T^{(j)})$. Достаточно сравнивать значения скалярных произведений $(g, T^{(j)} - T^{(0)})$. Координаты вершин $T^{(j)} - T^{(0)}$ записаны в явном виде.

Множество $M \downarrow_C$ представляет собой выпуклый многогранник в R^n , вершины которого $T^{(j)}$ ($j = 0, 1, \dots, n$) можно записать в явном виде:

$$T^{(0)} = 0; \quad T_i^{(j)} = \begin{cases} C, & i \leq j, \\ 0, & i > j, \end{cases} \quad j = 1, 2, \dots, n.$$

Множество $\tilde{M} \downarrow_C$ представляет собой выпуклый многогранник в R^n , вершины которого $T^{(j)}$ ($j = 0, 1, \dots, n$) можно записать в явном

виде следующим образом:

$$T^{(0)} = 0; \quad T_i^{(j)} = \begin{cases} C, & i \leq j, \\ \frac{n-i}{n-j} C, & i > j, \quad j = 1, 2, \dots, n. \end{cases}$$

Множество $\tilde{M}_C \subset R^n$ представляет собой выпуклый многогранник в R^n , вершины которого $T^{(i,j)}$ $0 \leq i \leq j \leq n$ можно записать в явном виде:

$$T^{(0,0)} = 0; \quad T_k^{(i,j)} = \begin{cases} \frac{k-1}{i-1} C, & k < i, \\ C, & i \leq k \leq j, \\ \frac{n-k}{n-j} C, & k > j, \quad 1 \leq i \leq j \leq n. \end{cases}$$

2.6.3. Интерфейс к программе Compact

Программа `Zadanie2` (рис. 2.10) демонстрирует интерфейс доступа к программе `Compact`. На рис. 2.10 приведен листинг программы в *m*-кодах Матлаб, который был использован для выполнения расчетного задания. Отметим два важных обстоятельства. Во-первых, строки программного кода на рис. 2.10 пронумерованы только для удобства обсуждения, но сами номера строк не являются частью программного кода. Во-вторых, любой текст в промежутке между знаком % и концом текущей строки трактуется как комментарий и не является частью программного кода Матлаб.

Строки 1–2. Это заголовок программы-функции `Zadanie2`. Четыре константы (`Nval`, `C1`, `C2`, `IC`) объявлены глобальными, т. е. они видимы (достижимы) и могут быть изменены из любой функции, участвующей в работе программы `Zadanie2`.

Строки 4–6. Здесь заданы исходные данные для практического расчета: параметры `Aval`, `Bval`, `Cval`, `Dval`, соответствующие границам интервалов задания переменных $s \in [a, b]$ и $t \in [c, d]$, а также параметры `Nval` и `Mval`, которые соответствуют значениям n и m , определяющим количество узлов сеток по переменным s и t , соответственно.

Строки 10 и 11. Выбор одного из специальных множеств (параметр `IC`), а также задание констант `C1` и `C2`, ограничивающих функции этих множеств.

2. Регуляризация на компактных множествах

```
%-----
01 function Zadanie2()
02 global Nval C1 C2 IC
03 % Входные параметры задачи
04 Aval = 0; Bval = 1; % Диапазон переменной s
05 Cval = -2; Dval = 2; % Диапазон переменной t
06 Nval = 41; Mval = 41; % Кол-во узлов сетки по s и t
07 % Параметры специального множества: IC= '1' - монотонные функции;
08 % '2' - монотонно-выпуклые функции; '3' - выпуклые функции;
09 % '-1' - функции с вариацией, ограниченной 2*C2
10 IC = 2; % Выбор множества специальной структуры
11 C1 = 0; C2 = 1; % Константы, ограничивающие специальные множества
12 % Уровни выхода: по невязке, квадр. градиента, количеству итераций:
13 DL = 0; ANGRD = 0; Imax = 400;
14 % Конечно-разностная аппроксимация
15 hs = (Bval - Aval)/(Nval - 1); % Шаг по переменной s
16 ht = (Dval - Cval)/(Mval - 1); % Шаг по переменной t
17 Sval = (Aval:hs:Bval)'; % Сетка по переменной s
18 Tval = (Cval:ht:Dval)'; % Сетка по переменной t
19 Amat = amat(@yadro,Tval,Sval); % Матрица оператора ИУ
20 % Расчет точного решения и точной правой части
21 Z0 = zeros(Nval,1); % Допустимая стартовая точка 0
22 Zbar = zeros(Nval,1); % Задать массив для точного решения
23 Zbar = zbar(Sval); % Точное решение Zbar
24 Ubar = Amat*Zbar; % Точная правая часть Ubar
25 % Получение "возмущенной" правой части
26 Err_delta = 0.015; % Уровень "возмущения" Ubar -> plus/minus 1.5%
27 Udelta = urandom(Ubar,delta); % "Возмущение" правой части ИУ
28 uval = Udelta; % Назначение функции для использования в расчете
29 [Za,AN,ITER,IERR] = compact(Amat,Z0,uval,(DL/ht),ANGRD,Imax);
30 AN = AN*ht; % Невязка
31 DU = Ubar - Udelta;
32 delta2 = (DU'*DU)/(Mval - 1); % Квадрат ср.кв.погрешности
33 % Выводим расчетные данные и графики
34 Zlim = [0 1.05]; % Диапазон изменения z(s) на графике
35 Ulim = [0 0.25]; % Диапазон изменения u(t) на графике
36 Ut = integral(Tval,Aval,Bval); % Аналитическая формула для интеграла
37 printout2; % <<<<<<<==== Процедура печати результатов для ИД3-2
38 % Анализатор константы C2 в диапазоне от Ca до Cb с шагом Nstep
39 Ca = 0.5; % Нижний предел диапазона сканирования C2
40 Cb = 2; % Верхний предел диапазона сканирования C2
41 Nstep =100; % Количество точек сканирования константы C2
42 Cmin = 0.9; % Левая граница оси C2 на графике
43 Cmax = 2.0; % Правая граница оси C2 на графике
44 Fmin = 0.5e-5; % Нижняя граница невязки на графике
45 Fmax = 2e-5; % Верхняя граница невязки на графике
46 c2_analyser; % <<<<<===== Процедура сканирования константы C2
47 end
%-----
```

Рис. 2.10. Листинг программы Zadanie2 в *m*-кодах Матлаб

Строка 13. Спецификация уровней выхода из итерационного процесса: по невязке (DL), квадрату градиента (ANGRD), максимальному количеству итераций (Imax).

Строки 15–16. Расчет величины шага и узлов равномерных сеток по переменным s и t .

Строки 17–18. Расчет сеток по переменным s (Sval) и t (Tval).

Строка 19. Расчет матрицы Amat для аппроксимации ядра ИУ:

$$\text{Amat} \equiv [A_{ij}] = \left[\frac{h_s}{1 + \delta_{1j} + \delta_{Nj}} K(t_i, s_j) \right],$$

где δ_{kj} – дельта-символ Кронекера. Интегрирование здесь будет по второй переменной – s . Так, что

$$U = \text{Amat} \times Z.$$

Для получения матрицы оператора Amat использована вспомогательная программа-функция `amat`, тремя входными параметрами которой являются: `yadro` – стандартное имя функции ядра $K(t, s)$ интегрального уравнения; Tval и Sval – сетки по переменным t и s .

Функция `yadro()` определяется пользователем в зависимости от аналитического выражения для ядра интегрального уравнения $K(t, s)$. На рис. 2.11 приведен пример оформления функции `yadro()` для ядра интегрального уравнения (2.20).

```
%.....
% yadro - определяемая пользователем функция ядра K(t,s)
% Функция используется программой Zadanie2
% Место расположения: отдельный файл yadro.m
%.....
01 function Kts = yadro(t,s)      % Модельная функция ядра для расчета
02 Kts = 1./(1 +100*(s - t).^2); % Аналитическая формула ядра
03 return;                      % Возврат из подпрограммы-функции
04 end                          % Конец подпрограммы
%.....
```

Рис. 2.11. Листинг программы `yadro` в m -кодах Матлаб

Строки 21–24. Получение точных значений решения (Zbar) и правой части (Ubar) уравнения по заданному значению точного решения \bar{z} , оформленному в виде функции `zbar()`, размещенной в файле `zbar.m`.

Строки 26, 27. Получение «возмущенной» правой части. Для этого использовали вспомогательную программу-функцию `urandom`, входными параметрами которой являются: имя переменной для внесения «возмущения» и величина этого возмущения `Err_delta`.

Строка 28. Функция `uval`, используемая в качестве правой части в расчете приближенного решения, определяется через `Ubar` или `Udelta`.

Строка 29. Вызов процедуры `compact` для решения задачи. Выходные данные процедуры: `Za` – приближенное решение, `AN` – величина невязки (необходимо учесть множитель `ht`), `ITER` – количество выполненных итераций, `IERR` – код завершения итераций.

Строки 30–32. Расчет квадрата среднеквадратической погрешности правой части $u(t)$ после внесения «возмущения».

Строки 34, 35. Задание диапазонов переменных для вывода графиков в процедуре `printout2`.

Строка 36. Расчет интеграла U_t по аналитической формуле оформляют в виде процедуры, размещенной в файле `integral.m`.

Строка 37. Вызов процедуры форматного вывода результатов, реализованной в виде отдельного файла `printout2.p`.

Строки 39–45. Задание параметров для работы анализатора константы `C2` (`c2_analyser`), осуществляющего расчет невязки в зависимости от значения константы `C2`.

Строка 46. Вызов процедуры анализатора `c2_analyser`, расчет и вывод графика зависимости невязки от значения константы `C2`.

Строка 47. Завершение программы `Zadanie2`.

2.6.4. Вывод результатов тестового расчета

Листинг программы `Zadanie2`, представленный на рис. 2.10, содержит пример для тестового расчета. В качестве модельной задачи для тестового расчета в примере выбрано интегральное уравнение (2.10)

$$\int_a^b K(t, s) z(s) \, ds = u(t), \quad t \in [c, d]$$

с гладким ядром

$$K(t, s) = \frac{1}{1 + 100(t - s)^2}$$

и значениями пределов интегрирования $a = 0$, $b = 1$, $c = -1$, $d = 2$.

Пользователь определяет три функции: `yadro` (ядро $K(t, s)$ интегрального уравнения), `integral` (аналитическая формула интеграла $\hat{A}\bar{z}$) и `zbar` (точное решение \bar{z}). Каждая из них оформлена в виде подпрограммы и размещена в одноименном файле с расширением `*.m`, который находится в рабочей директории пользователя, т. е. там же, где `Zadanie2.m` и `Compact.m`. Пример оформления функции `yadro`

показан на рис. 2.11. Функция `zbar` служит для расчета точной правой части `Ubar` уравнения. В тестовом примере задана функция `zbar` следующего вида:

$$\bar{z}(s) = 1 - s^2.$$

Аналитическое выражение (`Ut`) для точной правой части получают при аналитическом вычислении интеграла (2.10) и оформляют в виде отдельной подпрограммы `integral()`, которую размещают в рабочей директории пользователя в виде файла `integral.m`.

Множество монотонно выпуклых функций (`IC=2`), ограниченных константами `C1=0` и `C2=1` было выбрано в качестве множества специальной структуры (компакта) для поиска приближенного решения интегрального уравнения (2.10).

Для получения «возмущенной» правой части (`Udelta`) интегрального уравнения выбрано значение параметра `delta=0.015`, соответствующее уровню вносимого «возмущения» $\delta = \pm 1.5\%$.

При решении уравнения (2.10) правая часть уравнения в программе должна быть представлена функцией `uval`, которая может соответствовать одной из уже рассчитанных функций: `Ut`, `Ubar` или `Udelta`. В тестовом примере выполнено назначение `uval=Udelta`.

Для расчета приближенного решения (`Za`) в тестовом примере была использована процедура `compact`, а для вывода результатов расчета применяли вспомогательную процедуру `printout2`, оформленную в виде соответствующего скрипт-файла `printout2.p`.

На рис. 2.12–2.14 показан вывод на экран панели Матлаб результатов расчета модельной задачи, проведенный с использованием программного обеспечения `Zadanie2`, в котором использованы процедуры: `compact` – для расчета; `printout2` и `c2_analyser` – для текстового и графического вывода результатов расчета.

Приведенные экранные формы включают в себя диаграммы «Решение» (см. рис. 2.12), «Правая часть уравнения» (см. рис. 2.13) и «Оптимизация константы C » (см. рис. 2.14). Табличный вывод точного и приближенного решений включает в себя табулированные значения точного решения \bar{z} (`Zbar`), приближенного решения z_δ (`Za`) и некоторых параметров расчета.

2.7. Выполнение численного расчета в среде Матлаб

Цель проведения численного расчета: используя алгоритм регуляризации на компактных множествах, найти приближенное решение, удовлетворяющее интегральному уравнению Фредгольма 1-го рода с гладким ядром.

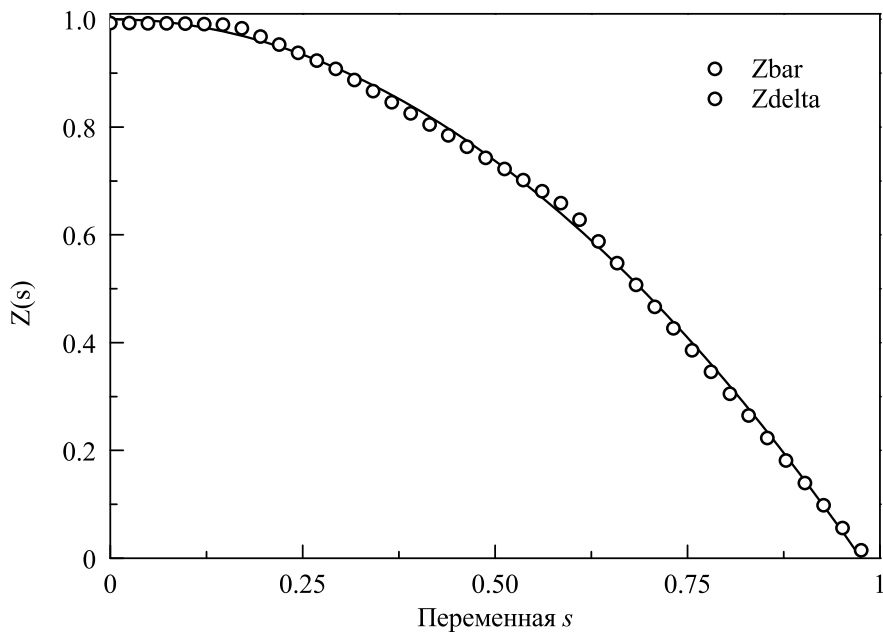


Рис. 2.12. Диаграмма «Решение» при выводе на экран результатов расчета модельной задачи

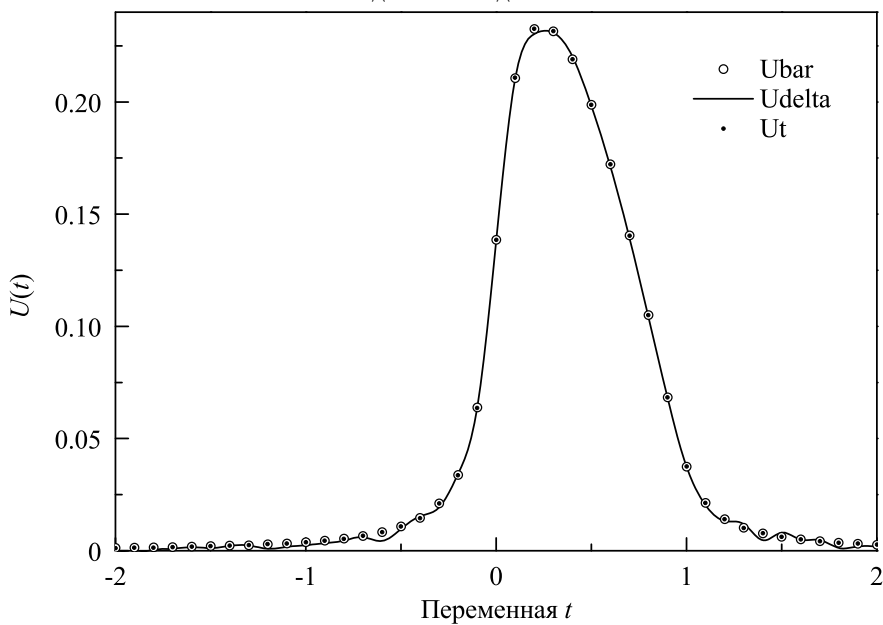


Рис. 2.13. Диаграмма «Правая часть» при выводе на экран результатов расчета модельной задачи

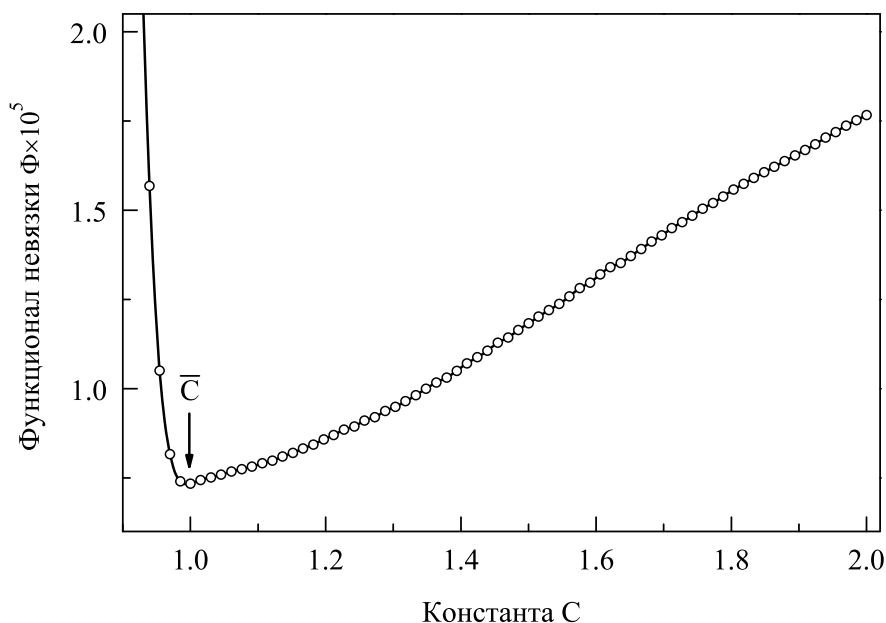


Рис. 2.14. Диаграмма «Оптимизация константы C» при выводе на экран результатов расчета модельной задачи

1. Получив индивидуальное задание, необходимо рассчитать точное значение правой части уравнения (2.1), т. е. функцию

$$\bar{u}(t) = \int_a^b K(t, s) \bar{z}(s) ds.$$

2. Составить определяемые пользователем функции ядра (**yadro**), точного решения (**zbar**) и интеграла (**integral**), которые оформляются в виде отдельных файлов **yadro.m**, **zbar.m** и **integral.m** (см. рис. 2.11). Эти три файла следует разместить в рабочей директории среды Матлаб. Если в этой директории уже присутствуют одноименные старые файлы, то их следует заменить на новые файлы, составленные пользователем в соответствии с заданием.

3. Внести в шаблон функции **Zadanie2** (см. рис. 2.10) данные своего индивидуального задания:

- диапазон $[a, b]$ изменения переменной s (строка 04);
- диапазон $[c, d]$ изменения переменной t (строка 05);

4. Задать и внести в шаблон функции **Zadanie2** (рис. 2.10) варьируемые параметры расчетного задания:

- количество узлов сетки $Nval$ и $Mval$ (строка 06);
- константу $C2$ (строка 11), которая соответствует константе C , используемой в теоретических расчетах и ограничивающей компактное множество сверху;
- ограничение максимального (предельного) количества итераций I_{max} (строка 13);
- уровень возмущения правой части $Ubar$ (строка 26).

5. Выбрать и задать наиболее подходящее специальное множество функций (монотонные, монотонно выпуклые, выпуклые, ограниченной вариации) – параметр IC (строка 10). Если не удалось получить удовлетворительный результат, то необходимо повторить расчет, используя другое специальное множество и заново выполнить п.4.

6. Оптимизировать результат расчета путем подбора константы $C2$ (строка 11). Для этого необходимо провести серию расчетов при различных значениях константы $C2$, составить зависимость величины невязки от константы $C2$ (построить график) и выбрать значение $C2$, соответствующее минимальному значению невязки. Похожие примеры выбора константы $C2$ приведены на рис. 2.6 – 2.9 и рис. 2.14.

7. Зафиксировать численные результаты расчета для последующего использования в пояснительной записке.

8. Построить две графические диаграммы. Диаграмма «Решение» будет содержать функции $Zbar(s)$ и $Za(s)$ – решения, которые получаются при работе программы *Zadanie2*. Диаграмма «Правая часть уравнения» должна содержать функции: $Ut(t)$ – результат аналитического вычисления интеграла; $Ubar(t)$ и $Udelta(t)$ – результаты работы программы *Zadanie2*.

9. Написать обоснованные выводы, где отразить, в частности, следующие результаты выполнения задания:

- оптимальный выбор специального множества;
- оптимальное значение ограничивающей константы $C2$;
- минимальный достигнутый уровень функционала невязки;
- сравнение графиков точного ($Zbar$) и приближенного (Za) решений;
- сравнение трех графиков, соответствующих правой части $u(t)$;
- влияние вносимого уровня возмущения правой части уравнения на найденное приближенное решение.

10. Оформить результаты работы в виде пояснительной записки согласно установленным требованиям.

3. Регуляризация по методу обобщенной невязки

Фундаментальные основы этого метода, включающие в себя понятия *регуляризации* и *регуляризирующего оператора*, были разработаны еще в 1943 г. А. Н. Тихоновым [7]. Суть регуляризации решения заключается в нахождении набора обратных операторов, зависящих от параметра регуляризации α , каждый из которых соответствует решению корректной задачи так, что он будет стремиться к истинному решению \bar{z} исходной некорректной задачи при одновременном стремлении к нулю параметра регуляризации α и ошибки δ исходных данных.

Самое главное отличие метода регуляризации А. Н. Тихонова от других известных методов решения некорректно поставленных задач состоит в его применимости к решению задач, когда класс возможных решений уравнения не является *компактом*. Такая ситуация характерна для многих обратных задач физической диагностики, а также для задач восстановления изображений.

3.1. Метод регуляризации А.Н. Тихонова

1. **Постановка задачи.** Зададим Z и U – гильбертовы пространства решений и правой части интегрального уравнения; D – замкнутое выпуклое множество ($D \subseteq Z$, $0 \in D$), соответствующее априорно известным ограничениям задачи; \hat{A} и \hat{A}_h – линейные ограниченные операторы, где \hat{A}_h – аппроксимирующий оператор, $h \geq 0$ – погрешность аппроксимации оператора, т. е. $\|\hat{A} - \hat{A}_h\| \leq h$.

Решение уравнения $\hat{A}z = u$ принадлежит множеству D , с учетом этого построим приближенное решение по заданному набору данных

$$\{\hat{A}_h, u_\delta, \eta\}, \quad \eta = (\delta, h),$$

где $\delta > 0$ такое, что $\|u - u_\delta\| \leq \delta$ – погрешность задания правой части уравнения. Запишем сглаживающий функционал (функционал Тихонова) с параметром регуляризации $\alpha > 0$:

$$M^\alpha(z) = \|\hat{A}_h z - u_\delta\|^2 + \alpha \Omega(z), \quad (3.1)$$

где $\Omega(z)$ – стабилизирующий функционал (стабилизатор) задачи, под которым понимают некоторый неотрицательный функционал, определенный на подмножестве $D \subseteq Z$ так, что для любого числа $C > 0$ множество D_C элементов $z \in D$, для которого $\Omega(z) \leq C$, является компактом на множестве D . Выбор функционала $\Omega(z)$ не является однозначным.

Пусть Z – множество функций $z(s)$, непрерывных на интервале $[a, b]$, а D – множество функций, имеющих производные до n -го порядка и интегрируемых с квадратом, тогда в качестве стабилизатора можно выбрать

$$\Omega(z) = \int_a^b \sum_{k=0}^n p_k(s) \left(\frac{d^k z}{d s^k} \right)^2 ds, \quad (3.2)$$

где $p_k(s)$ – заданные неотрицательные функции (весовые функции). Функционалы, задаваемые выражением (3.2), характеризуют «гладкость» функции $z(s)$ и носят общее название стабилизаторов n -го порядка.

Порядок n стабилизатора характеризует также степень регуляризации, которая может быть *слабой* ($n = 0$) или *сильной* ($n = 1$). При значениях $n > 1$ имеет место *гладкая* регуляризация порядка $(n - 1)$.

На практике рекомендуют использовать стабилизатор первого порядка ($n = 1$) с весовыми коэффициентами $p_0 = p_1 = 1$, а в качестве пространства Z выбирать пространство Соболева $W_2^1[a, b]$ с нормой

$$\|z\|^2 = \int_a^b \sum_{k=0}^n \left| z^{(k)}(s) \right|^2 ds, \quad (3.3)$$

где (k) обозначает производную k -го порядка. В этих условиях

$$\Omega(z) = \|z\|^2,$$

а сглаживающий функционал (функционал Тихонова) будет иметь вид

$$M^\alpha(z) = \|\hat{\mathbf{A}}_h z - u_\delta\|^2 + \alpha \|z\|^2. \quad (3.4)$$

Сформулируем экстремальную задачу минимизации функционала $M^\alpha(z)$:

$$z_\eta^\alpha = \inf_{z \in D} M^\alpha(z). \quad (3.5)$$

Для любых $\alpha > 0$, $u_\delta \in U$ и линейного оператора $\hat{\mathbf{A}}_h$ задача (3.5) разрешима и имеет единственное решение z_η^α , при этом выполняется неравенство $\|z_\eta^\alpha\| \leq \|u_\delta\| / \sqrt{\alpha}$.

Необходимым условием того, что z_η^α – точка минимума $M^\alpha(z)$ на множестве D , является условие $[M^\alpha(z_\eta^\alpha)]' = 0$, соответствующее уравнению Эйлера

$$\hat{\mathbf{A}}_h^* \hat{\mathbf{A}}_h z_\eta^\alpha + \alpha [z_\eta^\alpha - (z_\eta^\alpha)'] = \hat{\mathbf{A}}_h^* u_\delta, \quad (3.6)$$

где $\hat{\mathbf{A}}_h^*$ – оператор, сопряженный $\hat{\mathbf{A}}_h$. В таком случае вместо минимизации функционала (3.4) следует решать *уравнение Эйлера* (3.6).

2. Сопряженный оператор. Пусть $\hat{\mathbf{A}}$ – линейный ограниченный оператор, отображающий гильбертово пространство H_1 в гильбертово пространство H_2 . Оператор $\hat{\mathbf{A}}^*: H_2 \rightarrow H_1$ называется сопряженным к оператору $\hat{\mathbf{A}}$, если для всех $x \in H_1$, $y \in H_2$ выполняется равенство

$$(y, \hat{\mathbf{A}}x)_{H_2} = (\hat{\mathbf{A}}^* y, x)_{H_1}.$$

Оператор $\hat{\mathbf{A}}: H \rightarrow H$, совпадающий со своим сопряжением, называется *самосопряженным оператором*.

З а м е ч а н и е. Если линейный непрерывный оператор $\hat{\mathbf{A}}$ определен не на всем пространстве H_1 , то у него может быть несколько сопряженных операторов.

Если линейный оператор $\hat{\mathbf{A}}$ непрерывен, то $\hat{\mathbf{A}}^*$ тоже непрерывен, причем $\|\hat{\mathbf{A}}^*\| = \|\hat{\mathbf{A}}\|$. Отметим общие свойства сопряженных операторов:

1. $\|\hat{\mathbf{A}}^* \hat{\mathbf{A}}\| = \|\hat{\mathbf{A}} \hat{\mathbf{A}}^*\| = \|\hat{\mathbf{A}}\|^2$;
2. $(\|\hat{\mathbf{A}}^*\|)^* = \hat{\mathbf{A}}$;
3. $(\alpha \hat{\mathbf{A}} + \beta \hat{\mathbf{A}})^* = \alpha \hat{\mathbf{A}}^* + \beta \hat{\mathbf{A}}^*$;
4. $(\hat{\mathbf{A}} \hat{\mathbf{B}})^* = \hat{\mathbf{B}}^* \hat{\mathbf{A}}^*$.

Если $\hat{\mathbf{A}}$ – вполне непрерывный оператор, то тем же свойством обладает сопряженный оператор $\hat{\mathbf{A}}^*$.

Приведем пример сопряженного оператора. Пусть непрерывный оператор $\hat{\mathbf{A}} \{L_2[a, b] \rightarrow L_2[a, b], z \in H_1, u \in H_2\}$ задан формулой

$$\hat{\mathbf{A}}z = \int_a^b K(t, s) z(s) ds.$$

Сопряженный оператор $\hat{\mathbf{A}}^*$ определяется из условия

$$(u, \hat{\mathbf{A}} z)_{H_2} = (\hat{\mathbf{A}}^* u, z)_{H_1}$$

для любых $z, u \in L_2[a, b]$, следовательно

$$\begin{aligned} (u, \hat{\mathbf{A}} z)_{L_2} &= \int_a^b u(t) \left[\int_a^b K(t, s) z(s) ds \right] dt = \\ &= \int_a^b z(s) \left[\int_a^b K(t, s) u(t) dt \right] ds = \\ &= \int_a^b \left[\int_a^b K^*(t, \tau) u(\tau) d\tau \right] z(t) dt = (\hat{\mathbf{A}}^* u, z)_{L_2}, \end{aligned}$$

где $\hat{\mathbf{A}}^* u = \int_a^b K^*(t, \tau) u(\tau) d\tau$, $K^*(t, \tau) = K(\tau, t)$. В евклидовом пространстве, где линейные ограниченные операторы имеют матричную структуру $[A] = [a_{i,j}]_{m \times n}$, сопряженным операторам в ортонормированном базисе отвечают *сопряженные матрицы* $[A^*] = [a_{i,j}^*]_{n \times m}$, где $a_{i,j}^* = \bar{a}_{j,i}$. Если матрица $[A]$ – вещественная, то $[A^*] = [A^T]$.

Подытожим формулы для вычисления сопряженного оператора $\hat{\mathbf{A}}^*$:

$$\hat{\mathbf{A}}^* u = \int_a^b K^*(s, t) u(t) dt = \int_a^b K(t, s) u(t) dt. \quad (3.7)$$

3. Выбор параметра регуляризации. Выбор параметра регуляризации осуществляется в соответствии с принципом обобщенной невязки, т. е. параметр α находят из уравнения

$$\rho(\alpha) \equiv \| \hat{\mathbf{A}}_h z_\eta^\alpha - u_\delta \|^2 - (\delta + h \| z_\eta^\alpha \|^2 - (\mu(u_\delta, \hat{\mathbf{A}}_h))^2 = 0, \quad (3.8)$$

где $\mu(u_\delta, \hat{\mathbf{A}}_h) = \inf_{z \in D} \| \hat{\mathbf{A}}_h z - u_\delta \|$ – мера несовместности исходного уравнения с приближенными данными.

Если выполнено условие $\| u_\delta \|^2 \geq \delta^2 + (\mu(u_\delta, \hat{\mathbf{A}}_h))^2$, то уравнение (3.8) имеет один положительный корень, который выбирается в качестве параметра регуляризации в методе А. Н. Тихонова.

В упрощенном виде принцип обобщенной невязки формулируется в предположении, что $\mu(u_\delta, \hat{\mathbf{A}}_h) = 0$, тогда алгоритм принципа обобщенной невязки выглядит следующим образом.

Основа алгоритма – проверка условия $\|u_\delta\| > \delta$.

1. Если это условие не выполнено, то полагаем $z_\eta = 0$.

2. Если это условие выполнено, то:

a) если найдется α^* – корень функции ρ_η^α , то в качестве решения выберем $z_\eta^{\alpha^*} > 0$;

b) если $\rho_\eta^\alpha > 0$ для всех $\alpha > 0$, то положим приближенное решение равным $z_\eta = \lim_{\alpha \rightarrow 0+0} z_\eta^\alpha$.

4. Отыскание корня уравнения. Для вычисления корня уравнения (3.8) можно использовать любой подходящий численный метод, который обычно реализован в используемых пакетах прикладных программ (Excel, Mathcad, Матлаб и др.).

При самостоятельной реализации численного метода минимизации функционала обобщенной невязки (3.8) рекомендуется использовать модификацию *метода хорд*. На рис. 3.1 показан результат применения итерационной процедуры нахождения параметра регуляризации α с использованием модифицированного метода хорд.

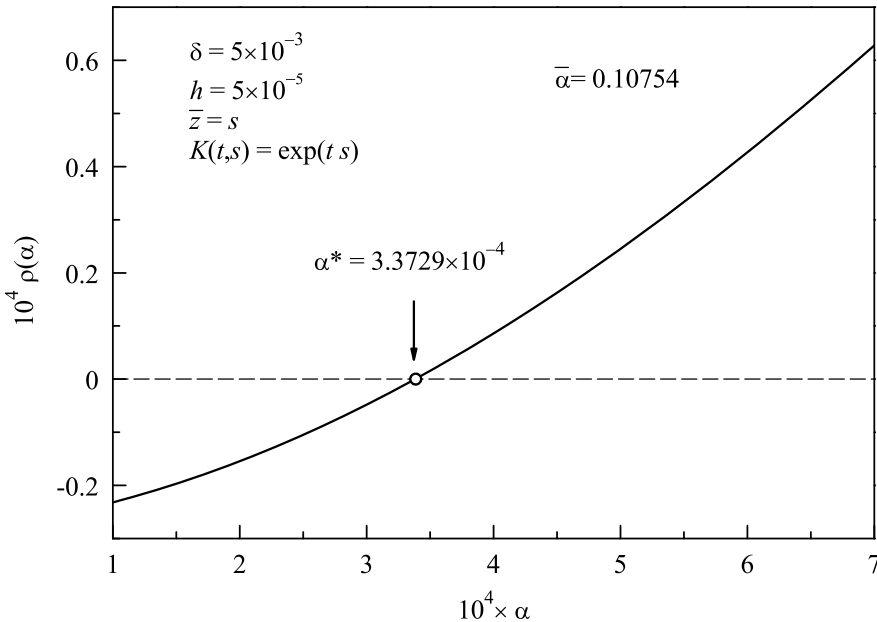


Рис. 3.1. Результат расчета корня уравнения $\rho(\alpha^*) = 0$, найденный за 13 итераций модифицированным методом хорд

Пусть задан параметр точности расчета $\varepsilon > 0$. Зададим начальное значение параметра α_0 и выберем следующее значение α_1 ($\alpha_1 < \alpha_0$, $|\alpha_1 - \alpha_0| \geq \varepsilon$), полагая $\alpha_1 = \alpha_0/2$ и вычислим соответствующие значения функций $\rho(\alpha_0)$, $\rho(\alpha_1)$.

Далее, до тех пор пока выполняется условие $|\alpha_{n-2} - \alpha_{n-1}| \geq \varepsilon$, строим итерационную последовательность, используя рекуррентную формулу

$$\alpha_n = \frac{\alpha_{n-2}}{1 - \left(1 - \frac{\alpha_{n-2}}{\alpha_{n-1}}\right) \left(\frac{\rho(\alpha_{n-2})}{\rho(\alpha_{n-2}) - \rho(\alpha_{n-1})}\right)}. \quad (3.9)$$

если $\rho(\alpha_{n-2}) \rho(\alpha_{n-1}) > 0$, то $\alpha_{n-2} = \alpha_{n-1}$, $\alpha_{n-1} = \alpha_n$;

если $\rho(\alpha_{n-2}) \rho(\alpha_{n-1}) < 0$, то $\begin{cases} \text{если } \rho(\alpha_{n-2}) \rho(\alpha_n) < 0, & \alpha_{n-1} = \alpha_n, \\ \text{если } \rho(\alpha_{n-1}) \rho(\alpha_n) < 0, & \alpha_{n-2} = \alpha_n. \end{cases}$

В качестве значения параметра регуляризации выбирают $\alpha = \alpha_{n-1}$.

3.2. Регуляризация для уравнения Фредгольма 1-го рода с гладким ядром

1. Конечно-разностная аппроксимация. В качестве некорректно поставленной задачи рассмотрим решение интегрального уравнения Фредгольма 1-го рода с гладким ядром.

$$\hat{\mathbf{A}} z(s) \equiv \int_a^b K(t, s) z(s) ds = u(t), \quad t \in [c, d], \quad s \in [a, b]. \quad (3.10)$$

Определим, что $K(t, s) \in C([c, d] \times [a, b])$, $z \in C[a, b]$, $u \in C[c, d]$.

Пусть существует априорная информация о том, что $z(s)$ – кусочно-гладкая функция, тогда выберем пространство $Z = W_2^1[a, b]$. В пространстве Соболева $W_2^1[a, b]$ норма определена следующим образом:

$$\|z\|_{W_2^1[a, b]}^2 = \int_a^b \left[(z(s))^2 + (z'(s))^2 \right] ds. \quad (3.11)$$

В качестве другого пространства выберем пространство $U = L_2$, где вместо u известно приближенное значение u_δ такое, что $\|u - u_\delta\| \leq \delta$, а вместо $K(t, s)$ известна такая функция $K_h(t, s)$, что $\|K - K_h\| \leq h$,

тогда $\|\hat{\mathbf{A}} - \hat{\mathbf{A}}_h\| \leq h$, где $\hat{\mathbf{A}}_h$ – интегральный оператор, соответствующий ядру $K_h(t, s)$. Определение нормы в пространстве L_2 имеет вид:

$$\|u\|_{L_2[c,d]}^2 = \int_c^d (u(t))^2 dt. \quad (3.12)$$

Перейдем от уравнения (3.10) к минимизации функционала $M^\alpha(z)$

$$\begin{aligned} M^\alpha(z) &= \|\hat{\mathbf{A}}_h z - u_\delta\|_{L_2[c,d]}^2 + \alpha \|z\|_{W_2^1[a,b]}^2 = \\ &= \int_c^d \left[\int_a^b K_h(t, s) z(s) ds - u_\delta(t) \right]^2 dt + \alpha \int_a^b [(z(s))^2 + (z'(s))^2] ds. \end{aligned} \quad (3.13)$$

Необходимым условием того, что z_η^α – точка минимума $M^\alpha(z)$ на D , является условие $[M^\alpha(z_\eta^\alpha)]' = 0$, соответствующее уравнению Эйлера (3.6):

$$\int_a^b Q(s, s') z(s') ds' + \alpha [z(s) - z''(s)] = G(s), \quad s' \in [a, b], \quad (3.14)$$

где использованы условные обозначения $Q(s, s')$ и $G(s)$, соответствующие интегральным соотношениям, определенным следующим образом:

$$Q(s, s') \equiv \hat{\mathbf{A}}_h^* \hat{\mathbf{A}}_h = \int_c^d K(\tau, s) K(\tau, s') d\tau, \quad \tau \in [c, d];$$

$$G(s) \equiv \hat{\mathbf{A}}_h^* u_\delta = \int_c^d K(\tau, s) u_\delta(\tau) d\tau.$$

Для разрешимости уравнения Эйлера (3.14) требуется выполнение

$$p_1(s) \frac{dz}{ds} \delta z \Big|_a^b = 0, \quad (3.15)$$

где δz – произвольная вариация функции $z(s)$ такая, что $z(s) + \delta z(s)$ принадлежит классу допустимых функций. При наличии информации о значениях искомого решения $z(s)$ уравнения (3.10) на обоих концах

интервала $[a, b]$ допустимыми функциями при нахождения минимума функционала (3.13) будут лишь такие $z(s)$ из множества функций, интегрируемых с квадратом и имеющих обобщенную первую производную, которые принимают заданные значения в точках a и b . Для таких функций условие (3.15) должно заведомо выполняться, а значения δz будут обращаться в нуль при $s = a, s = b$.

Задача определения регуляризованного решения $z^\alpha(s)$ при этом сводится к решению интегродифференциального уравнения (3.14) с краевыми условиями

$$z(a) = z_1 \quad \text{и} \quad z(b) = z_2,$$

где z_1 и z_2 – известные числа. Если не известны значения искомого решения на концах интервала $[a, b]$, то условие (3.15) можно выполнить, записав $z'(a) = z'(b) = 0$. Возможны также другие формулировки краевых условий для уравнения (3.14).

2. Схема численного расчета. Уравнение (3.14) с заданными краевыми условиями может быть решено каким-либо подходящим численным методом. При численном решении это уравнение заменяется его конечно-разностной аппроксимацией на заданной сетке. Рассмотрим процесс аппроксимации более подробно.

Методика конечномерной аппроксимации функционала $M^\alpha(z)$ заключается в следующем. Зададим n и m – количество узлов по переменным s и t соответственно. Используя выбранные квадратурные формулы, построим равномерные сетки по переменным s и t с постоянными шагами

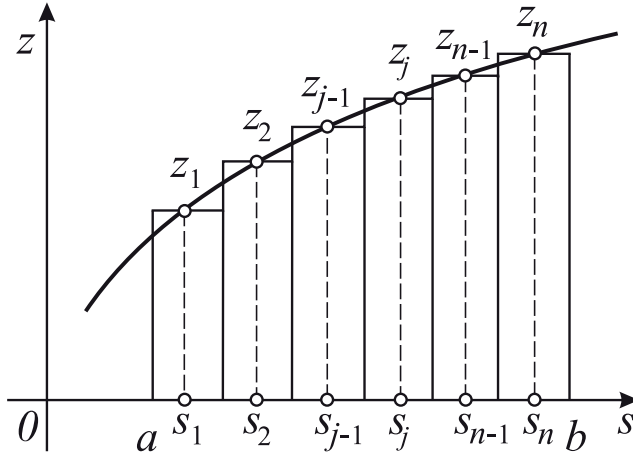
$$h_s = \frac{b - a}{n}, \quad h_t = \frac{d - c}{m},$$

а затем рассчитаем координаты узлов для обеих сеток по формулам

$$s_j = a + (j - 1/2) h_s, \quad j = \overline{1, n}; \quad t_i = c + (i - 1/2) h_t \quad i = \overline{1, m}.$$

На рис. 3.2 схематически показано построение сетки по переменной s и сеточная аппроксимация функции $z(s)$. Для переменной t и функции $u(t)$ сеточная аппроксимация выполнена аналогичным образом. Значение интеграла функции $z(s)$ в пределах $s \in [a, b]$ при численном интегрировании по методу прямоугольников приближенно аппроксимируется интегральной суммой

$$\int_a^b z(s) ds \approx h_s \sum_{j=1}^n z_j.$$


 Рис. 3.2. Сеточная аппроксимация функции $z(s)$

Обозначая $z(s_j) = z_j$, $u(t_i) = u_i$, $K(t_i, s_j) = a_{ij}$, используем квадратурную формулу прямоугольников для вычисления интегралов и аппроксимируем производную конечной разностью $z'(s) = (z_{j+1} - z_j)/h_s$. Тогда конечномерная аппроксимация функционала примет вид

$$M^\alpha(z_i) = h_t \sum_{i=1}^m \left[\sum_{j=1}^n h_s a_{ij} z_j - u_i \right]^2 + \alpha h_s \sum_{j=1}^n [(z_j)^2 + (z'_j)^2]. \quad (3.16)$$

Определим соответствующие матричные элементы следующим образом:

$G = [g_j]$ – вектор-столбец размерами $(n \times 1)$.

$$g_j = \int_c^d K(\tau, s_j) u_\delta(\tau) d\tau, \quad [g_j] \approx h_t \sum_{i=1}^m a_{ij} u_i, \quad G = h_t \times A^T U.$$

Здесь введены обозначения $A = [a_{ij}]$ – матрица размерами $(m \times n)$; функция $U = [u_i]$ – вектор-столбец размерами $(m \times 1)$.

$Q = [q_{jk}]$ – матрица размерами $(n \times n)$, элементы которой определены

$$q_{jk} = \int_c^d K(\tau, s_j) K(\tau, s_k) d\tau, \quad [q_{jk}] \approx h_t \sum_{i=1}^m (a_{ij})^T a_{ik}, \quad Q = h_t \times A^T A.$$

Используя необходимое условие минимума функционала, заменим в левой части уравнения (3.14) интеграл соответствующей ему интегральной суммой, например, по формуле прямоугольников, а вторую производную z'' – конечно-разностным отношением, получим

$$h_s \sum_{k=1}^n q_{jk} z_k + \alpha z_j + \frac{\alpha}{h_s^2} [2z_j - z_{j-1} - z_{j+1}] = g_j. \quad (3.17)$$

При $j = 1$ и $j = n$ в матрицу входят неопределенные еще значения z_0 и z_{n+1} . Для соблюдения краевых условий их принимают равными $z_0 = z_1$ и $z_{n+1} = z_n$.

Выполнив очевидные преобразования, имеем линейную алгебраическую систему с симметричной матрицей

$$B^\alpha z = U, \quad (3.18)$$

где использованы обозначения

$$B^\alpha = B + \alpha \left[E + \frac{1}{h_s^2} C \right], \quad (3.19)$$

здесь E – единичная матрица размером $(n \times n)$; $B = h_s \times Q$; матрица C – симметричная трехдиагональная матрица размерами $n \times n$. Ее крайние диагонали образованы числами (-1) , а главная диагональ состоит из чисел (2) . Исключение составляют крайние элементы главной диагонали: $c_{11} = 1$ и $c_{nn} = 1$

$$C = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & 1 \end{bmatrix} \quad (3.20)$$

Для решения системы линейных уравнений (3.18) можно использовать различные численные методы, например, метод Гаусса-Жордана. Однако при учете специфических свойств матрицы системы уравнений (симметричная и положительно определенная) возможно применение более эффективных методов решения такой системы, например, метод квадратного корня или метод разложения Холецкого.

3. Решение системы линейных уравнений. Рассмотрим теоретические предпосылки применения метода квадратного корня. Основная идея этого метода для заданной исходной квадратной матрицы (например, B^α) состоит в нахождении верхней треугольной матрицы T^α с неотрицательными диагональными элементами такой, что

$$B^\alpha = (T^\alpha)^T \times T^\alpha,$$

где

$$T^\alpha = \begin{bmatrix} t_{11}^\alpha & t_{12}^\alpha & \dots & t_{1n}^\alpha \\ 0 & t_{22}^\alpha & \dots & t_{2n}^\alpha \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & t_{nn}^\alpha \end{bmatrix}.$$

Для вычисления компонентов этой матрицы используются формулы:

$$t_{11}^\alpha = \sqrt{b_{11}^\alpha}, \quad t_{1j}^\alpha = \frac{b_{1j}^\alpha}{t_{11}^\alpha}, \quad (j > 1);$$

$$t_{ii}^\alpha = \left[b_{ii}^\alpha - \sum_{k=1}^{i-1} (t_{ki}^\alpha)^2 \right]^{1/2}, \quad (i = 2, \dots, n);$$

$$t_{ij}^\alpha = \left[b_{ij}^\alpha - \sum_{k=1}^{i-1} t_{ki}^\alpha t_{kj}^\alpha \right] \times \frac{1}{t_{ii}^\alpha}, \quad (i < j); \quad t_{ij}^\alpha = 0, \quad (i > j).$$

Известно [9], что для симметричной и положительно определенной исходной матрицы разложение *Холецкого* существует и притом единственно.

После определения компонентов матрицы $[T^\alpha]$ можно от системы уравнений (3.18) перейти к эквивалентной системе

$$(T^\alpha)^T \times T^\alpha z = U,$$

решение которой естественным образом разбивается на две более простые задачи с треугольными матрицами

$$\begin{cases} (T^\alpha)^T Y = U; \\ T^\alpha z = Y. \end{cases} \quad (3.21)$$

Решение системы с каждой из этих двух матриц может быть выполнено в Матлаб несколькими различными путями, например:

1) обращением матрицы путем применения операции `backslash` (обратная косая черта – решение путем обращения матрицы);

2) применением функции `linsolve`, основные преимущества которой над операцией `backslash` проявляются только при точной спецификации типа матрицы системы уравнений.

3.3. Регуляризация численного дифференцирования

Численное дифференцирование связано с конечно-разностной аппроксимацией задачи Коши для обыкновенных дифференциальных уравнений, которая в наших обозначениях имеет вид

$$z(t) = \frac{du(t)}{dt}, \quad u(a) = u_0, \quad a \leq t \leq b. \quad (3.22)$$

При численном решении поставленная задача является некорректной и требует регуляризации. Особенно сильно это сказывается при нахождении производных высокого порядка. Одним из способов регуляризации численного дифференцирования является регуляризация по методу обобщенной невязки. Рассмотрим основные этапы регуляризации численного дифференцирования.

Для начала задачу Коши (3.22) сводят к интегральному уравнению Вольтерра 1-го рода с единичным ядром:

$$\int_a^t z(s) ds = u(t) - u_0, \quad a \leq s \leq b. \quad (3.23)$$

Две различные по виду формулировки (3.22) и (3.23) являются полностью эквивалентными по своей сути, что легко проверяется прямой подстановкой выражения (3.22) в интегральное уравнение (3.23).

На следующем этапе уравнение (3.23) формально преобразуют к уравнению Фредгольма 1-го рода с разрывным (треугольным) ядром:

$$\begin{aligned} \int_a^b K(t, s) z(s) ds &= u(t) - u_0, & a \leq t \leq b; \\ K(t, s) &= 1 & \text{при } a \leq s \leq t \leq b, \\ K(t, s) &= 0 & \text{при } s > t. \end{aligned} \quad (3.24)$$

При выполнении данной формальной процедуры требуют, чтобы областью определения ядра $K(t, s)$ интегрального уравнения (3.24) был

квадрат $Q = [a, b] \times [a, b]$. Помимо этого, при конечно-разностной аппроксимации задачи (3.24) требуют, чтобы сетки по переменным t и s были одинаковы, т. е. должны выполняться условия: $m = n$, $h_s = h_t$.

Поскольку требование непрерывности ядра не является существенным, то к уравнению (3.24) полностью применим рассмотренный выше (см. п. 3.2) алгоритм регуляризации уравнения Фредгольма 1-го рода.

На рис. 3.3 представлен результат модельного расчета, выполненный для функции Gauss-Amp ($\delta = 10^{-5}$, $h = 10^{-5}$).

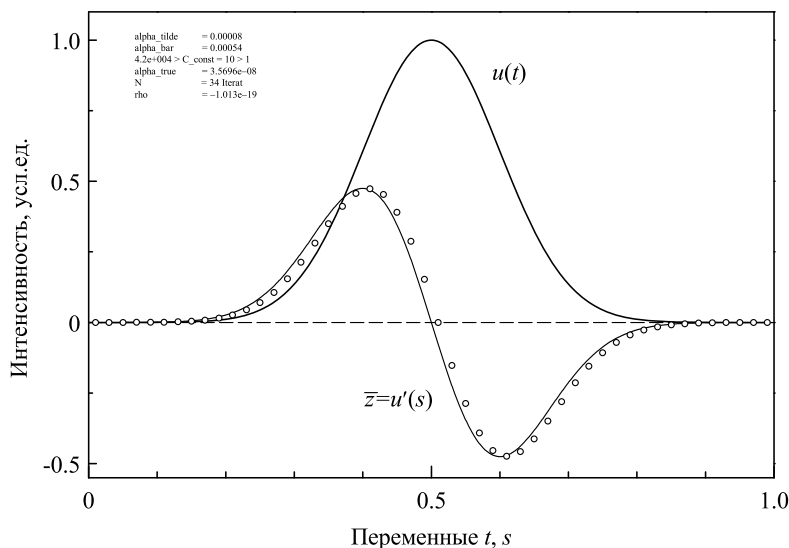


Рис. 3.3. Результат расчета задачи численного дифференцирования с регуляризацией по методу А. Н. Тихонова. Круги — приближенное решение z_δ

3.4. Программное обеспечение расчетов в среде Матлаб

Используя представленный выше регуляризирующий алгоритм для уравнения Фредгольма 1-го рода с гладким ядром, требуется найти приближенное решение, удовлетворяющее интегральному уравнению (3.10).

1. Алгоритм численного решения. Выберем следующие исходные данные для демонстрации алгоритма решения:

$$K(t, s) = \exp(t, s); \quad \bar{z}(s) = s; \quad a = 0; \quad b = 1; \quad c = 1; \quad d = 2.$$

Вычислим функцию $u(t)$ в правой части уравнения (3.10) (рис. 3.4).

При этом интеграл в левой части уравнения (3.10) может быть взят по частям:

$$\begin{aligned}
 u(t) &= \int_a^b K(t, s) \bar{z}(s) \, ds = \int_0^1 s \exp(t, s) \, ds = \\
 &= \left. \frac{s}{t} \exp(ts) \right|_0^1 - \frac{1}{t} \int_0^1 \exp(ts) \, ds = \\
 &= \frac{1}{t} \exp(t) - \frac{1}{t^2} \exp(ts) \Big|_0^1 = \frac{1}{t^2} [1 + (t - 1) \exp(t)] . \quad (3.25)
 \end{aligned}$$

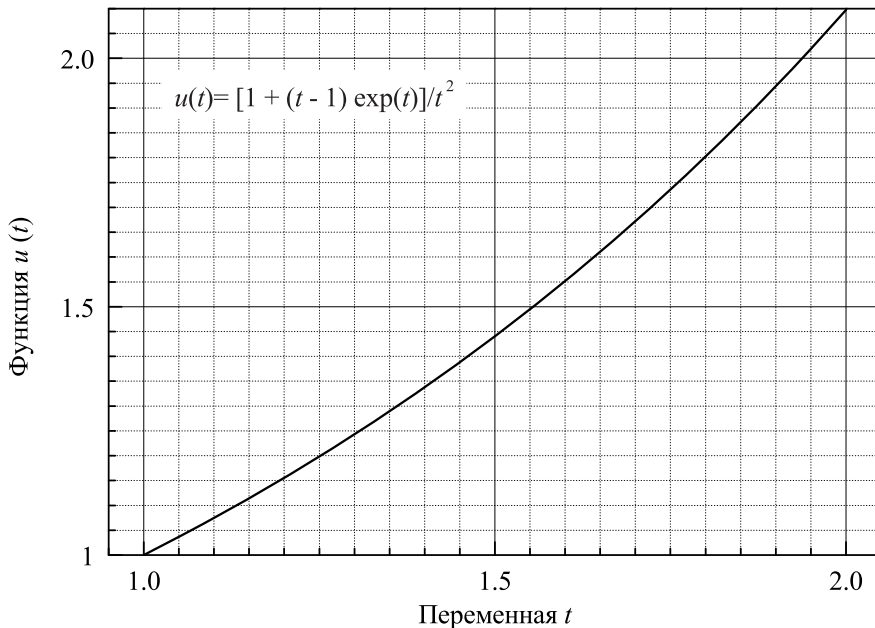


Рис. 3.4. Значения функции $u(t)$, рассчитанные на основании заданной $\bar{z}(s)$

Зададим сетки по переменным s и t . Для этого проведем разбивку отрезков $[c, d]$ и $[a, b]$, определив координаты соответствующих узлов t_i , $i = \overline{1, m}$; s_j , $j = \overline{1, n}$. Параметры разбивки выберем следующим образом: $m=20$; $n=50$. Этой разбивке соответствуют параметры шага $h_s = (b - a)/n = 0.02$; $h_t = (d - c)/m = 0.05$. Используя аналитическое выражение (3.25) для $u(t)$, полученное в п. 1, рассчитаем конечно-разностную аппроксимацию для $u_i = u(t_i)$, $i = \overline{1, m}$.

Зададим точность вычисления параметра регуляризации $\varepsilon = 1 \times 10^{-5}$, точность задания правой части $\delta = 5 \times 10^{-3}$ и погрешность аппроксимации оператора $h = 5 \times 10^{-5}$. Выберем вспомогательный параметр $C = 10$, рассчитаем $\tilde{\alpha} = 0.19782$, а затем вычислим оценку верхней границы параметра регуляризации $\bar{\alpha} = 0.10754$. Зададим стартовый параметр $\alpha_0 = \bar{\alpha}$ для поиска корня уравнения $\rho(\alpha) = 0$.

Используя процедуру *AlphaFinder*, получим значение параметра регуляризации $\alpha^* = 3.3729 \times 10^{-4}$.

Решая уравнение (3.18) для вычисленного α^* с помощью процедуры *AlphaFinder*, определим приближенное решение $z_{\eta}^{\alpha^*}$.

Построим графики решений \tilde{z} и $z_{\eta}^{\alpha^*}$ для уравнения (3.10), рис. 3.5.

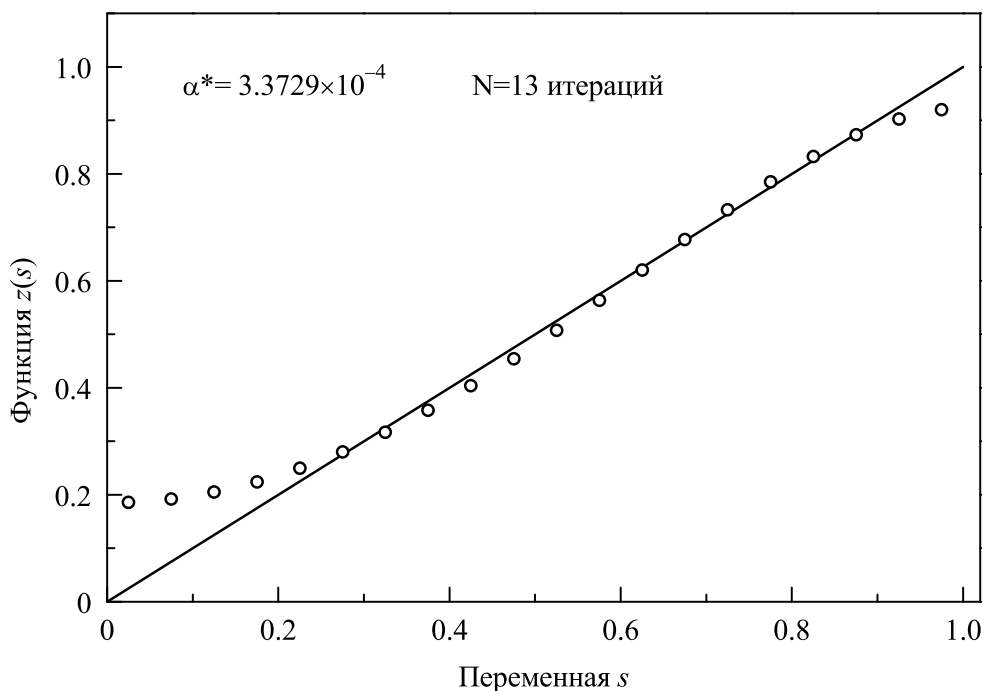


Рис. 3.5. Точное решение уравнения $\tilde{z}(s)$ (сплошная кривая) и приближенное решение $z_{\eta}^{\alpha^*}$, рассчитанное методом регуляризации А.Н. Тихонова, (кружки)

Зададим параметры `alpha0=0` и `Auto=0`, а затем с помощью процедуры *AlphaFinder* рассчитаем приближенное решение $z_{\eta}^{\alpha=0}$ (без регуляризации). Сравним два решения ($z_{\eta}^{\alpha=0}$ и $z_{\eta}^{\alpha^*}$).

Задавая ядро в виде треугольной матрицы (нижняя часть и главная диагональ – единицы, верхняя часть – нули), с помощью процедуры *AlphaFinder* выполним численное дифференцирование функции

правой части. Построим графики результатов численного и аналитического дифференцирования, сравним эти графики.

2. Программное обеспечение для выполнения расчетного задания. Для выполнения практических расчетов использовали программную среду Матлаб. На рис. 3.6 приведен листинг программы (*m*-коды Матлаб), который был использован для выполнения практического задания. Предваряя обсуждение программы, отметим два важных обстоятельства. Во-первых, строки программного кода на рис. 3.6 пронумерованы только для удобства обсуждения, но сами номера строк не являются частью программного кода. Во-вторых, любой текст в промежутке между знаком % и концом текущей строки трактуется как комментарий и не является частью программного кода.

Строки 1,2. Это заголовок программы-функции `Zadanie3()`. Две константы (`ACOR`, `SINGULAR`) и семь переменных (`Kmat`, `uval`, `hs`, `ht`, `Nval`, `Mval`, `eta`) объявлены глобальными, т.е. они видимы (достижимы) и могут быть изменены из любой функции, участвующей в работе программы `Zadanie3`.

Строки 4–7. Здесь заданы исходные данные для численного расчета: параметры `Aval`, `Bval`, `Cval`, `Dval`, соответствующие границам интервалов задания переменных $s \in [a, b]$ и $t \in [c, d]$; параметры `Err_delta` и `Err_h`, соответствующие ошибкам правой части (δ) и аппроксимации оператора (h).

Строки 9–14. Здесь заданы параметры для конечно-разностной аппроксимации решаемой задачи и настройки итерационного процесса: параметр `C_con`, соответствующий константе C (см. п. 3.4); `alpha0` – параметр, соответствующий стартовому значению для итерационного расчета параметра регуляризации α ; параметр `Err_alph`, соответствующий значению ε , используемому в модификации метода хорд для указания точности расчета α ; параметр `Imax`, задающий предельное количество итераций, после достижения которого происходит принудительное завершение итерационного процесса; параметр `Auto`, позволяющий включать (`Auto=1`) и выключать (`Auto=0`) автоматический выбор параметра регуляризации α , в качестве стартового значения которого используется `alpha0`; параметры `Nval` и `Mval` соответствуют значениям n и m , определяющим количество узлов сеток по переменным s и t , соответственно.

Строка 15. Вызов процедуры `taskini` (рис. 3.7) для начальной инициализации расчетной задачи, т.е. для расчета величины шагов `hs` (по переменной s), `ht` (по переменной t) и сеток `Sval` и `Tval`, соответствующих переменным s и t ; матрицы `Kmat` ядра интегрального уравнения; точных значений решения `Zbar` и правой части `Ubar`, возмущенной функции `Udelta`.


```

1 function Zadanie3() %-----
2 global ACOR SINGULAR Kmat uval hs ht Nval Mval eta
3 % Исходные данные для численного расчета
4 Aval = 0;    Bval = 1;                % Границы переменной s=[a, b]
5 Cval = 1;    Dval = 2;                % Границы переменной t=[c, d]
6 Err_delta = 5e-3;                    % delta - ошибка u(t)
7 Err_h      = 5e-5;                    % h - ошибка оператора
8 % Настройка параметров расчета
9 C_con      = 10;                      % Константа для расчета alpha0
10 alpha0     = 3e-4;                    % Стартовое значение параметра
11 Err_alph   = 5e-6;                    % Точность расчета alpha (epsilon)
12 Imax       = 50;                      % Предельное кол-во итераций
13 Auto       = 1;                      % Включить автоопределение alpha
14 Nval = 20;    Mval = 60;              % Кол-во узлов сетки по s (n) и t (m)
15 taskini;    % расчет hs, ht, Sval, Tval, Zbar, Kmat, Ubar, Udelta
16 uval = Udelta; % Использовать это значение uval в расчете
17 % Настройка параметров решения СЛАУ для min функционала Тихонова
18 ZSOLVE = 0; % Процедура 'linsolve' (0) или '\' (1)
19 SINGULAR = 0; % 0 - нет, 1 - есть сообщения о сингулярности
20 ACOR = 1e-12; % Параметр для корректировки сингулярности --> 0
21 % Определяем параметр регуляризации alpha
22 [za rho alpha cnt C1]=AlphaFinder(alpha0,C_con,Err_alph,Imax,Auto);
23 % Вывод расчетных данных и графиков на экран
24 Zlim = [0 1.05]; % Диапазон изменения z(s) на графике
25 Ulim = [1 2.2]; % Диапазон изменения u(t) на графике
26 datapoz = [0.05 0.95]; % Позиция текстовой информации [x0, y0]
27 Ut = integral(Tval,Aval,Bval); % Аналитическая формула интеграла
28 printout3; % <<<<<<<<<<==== Процедура печати результата
29 % Процедура численного дифференцирования на квадратной сетке [a,b]
30 Nval = 20; % Кол-во узлов сетки n = m
31 Mval = Nval; Cval = Aval; Dval = Bval; % Квадратная область [a,b]
32 taskini; % расчет hs, ht, Sval, Tval, Zbar, Kmat, Ubar, Udelta
33 uval = Udelta - Udelta(1); % Значение uval в расчете на новой сетке
34 % Корректировка сингулярности матрицы при вычислении alpha
35 ZSOLVE = 0; % Процедура 'linsolve' (0) или '\' (1)
36 SINGULAR = 0; % 0 - нет, 1 - есть сообщения о сингулярности
37 ACOR = 1e-12; % Параметр для корректировки сингулярности --> 0
38 Kmat = kmat(@yadrol,Tval,Sval); % Треугольное ядро
39 alph0 = 1e-6; % Стартовое значение параметра
40 Auto = 1; % Включить автоопределение alpha
41 Err_alph = 1e-7; % Точность расчета alpha (epsilon)
42 % Определяем параметр регуляризации alpha
43 [za rho alpha1 cnt C1]=AlphaFinder(alph0,C_con,Err_alph,Imax,Auto);
44 % Вывод расчетных данных и графиков на экран .....
45 Dlim = [0 1.3]; % Диапазон изменения du/dt на графике
46 Uscale = 1; % Коэффициент масштабирования u(t) в сравнении с du/dt
47 datapoz = [0.05 0.95]; % Позиция текстовой информации [x0, y0]
48 Ut = integral(Tval,Aval,Bval); % Аналитическая формула интеграла
49 Dudt = derivat(Ut,Sval,Tval); % Аналитическая производная dUt/dt
50 uplotter; % <<<<<<<<<<==== Процедура вывода du /dt
51 end %-----

```

Рис. 3.6. Листинг программы Zadanie3 в *m*-кодах Матлаб

Строка 16. Назначение функции `uval` для использования в расчете.

Строки 18–20. Настройка параметров решения СЛАУ для поиска минимума функционала Тихонова: `ZSOLVE` – переключение между процедурами `linsolve` (0) и `backslash` (1); `SINGULAR` – включение (1) корректировки сингулярности и настройка числового параметра `ACOR` для этой корректировки.

Строка 22. Вызов внешней процедуры `AlphaFinder`, которая осуществляет расчет в рамках метода регуляризации А.Н. Тихонова.

```
%=====
% taskini - скрипт-код для начальной инициализации задачи
% Вызов - из программы Zadanie3
% Вход:      Aval, Bval, Cval, Dval, Nval, Mval, Err_delta, Err_h
% Выход:     Zbar, Ubar, Kmat, Sval, Tval, eta, Udelta, hs, ht
%=====
1 % Готовим пустые массивы для последующего заполнения
2 Sval = zeros(Nval,1);          % s - вектор-столбец (n x 1)
3 Tval = zeros(Mval,1);          % t - вектор-столбец (m x 1)
4 Kmat = zeros(Mval,Nval);       % K(t,s) - матрица ядра ИУ (m x n)
5 Zbar = zeros(Nval,1);          % Za - приближенное решение (n x 1)
6 % Заполняем массивы и матрицы исходными данными
7 eta = [Err_delta Err_h];
8 hs = (Bval - Aval)/Nval; ht = (Dval - Cval)/Mval; % Шаги hs и ht
9 Tval = (Cval + ht/2):ht:(Dval - ht/2);
10 Sval = (Aval + hs/2):hs:(Bval - hs/2);
11 Zbar = zbar(Sval);            % Генерируем точное решение
12 % Генерирование правой части на сетке по t и s
13 Kmat = kmat(@yadro,Tval,Sval); % Матрица ядра ИУ на новой сетке
14 Ubar = hs*Kmat*Zbar;          % Точная правая часть (Ubar) на новой сетке
15 Udelta = urandom(Ubar,Err_delta); % "Возмущение" правой части ИУ
%=====
```

Рис. 3.7. Листинг программы `taskini` в *m*-кодах Матлаб

Входными параметрами процедуры являются: `alpha0` – стартовое значение для итерационного поиска корня уравнения $\rho(\alpha) = 0$; `C_const` – значение константы C для расчета верхней оценки параметра регуляризации; `Err_alpha` – соответствует параметру ε в модификации метода хорд; `Imax` – предельное значение количества итераций для принудительного завершения итерационного процесса; `Auto` – параметр, позволяющий включать (`Auto=1`) и выключать (`Auto=0`) автоматический выбор параметра регуляризации α . В последнем случае в качестве параметра регуляризации используется значение `alpha0`, заданное пользователем. Отметим также, что в качестве входных параметров процедура `AlphaFinder` использует также глобальные переменные, объявленные в строке 2 листинга программы.

Результатом выполнения процедуры являются следующие данные: **za** – значения функции приближенного решения $z_{\eta}^{\alpha*}$; **rho** – значения функции обобщенной невязки $\rho(\alpha^*)$; **alpha** – значение параметра регуляризации α^* , соответствующего корню уравнения $\rho(\alpha^*) = 0$; **cnt** – количество выполненных итераций, в результате которых было определено значение корня α^* ; **C1** – значение верхнего предела для константы C : $(\|u_{\delta}\|/\delta) > C > 1$.

Отметим, что процедура **AlphaFinder** оформлена в виде отдельного файла с именем **AlphaFinder.p**, который должен находиться в той же рабочей директории, где размещен файл **Zadanie3.m**.

Строки 24–26. Здесь размещены параметры для графического вывода данных на экран панели Матлаб.

Строка 27. Вызов процедуры для аналитического расчета функции **Ut** – значения интеграла в исходном интегральном уравнении. Аналитическая формула размещена в отдельном файле **integral.m**.

Строка 28. Вызов процедуры для вывода текстовой и графической информации – результатов расчета, на экран панели Матлаб.

На рис. 3.8 показан пример текстового вывода результатов расчета на экран панели Матлаб. Показаны результаты двух расчетов для одного и того же набора входных данных: слева – расчет без регуляризации ($\alpha=0$), справа – расчет с регуляризацией по методу А.Н. Тихонова ($\alpha=3.3729 \times 10^{-4}$). Значение параметра регуляризации было получено при помощи процедуры **AlphaFinder**. Параметр **alpha_true** соответствует α^* , **Za** соответствует $z_{\eta}^{\alpha*}$, **rho** соответствует $\rho(\alpha^*)$, значение **C1** = 302, количество итераций **N**=13 (см. рис. 3.8).

Вторая половина программы **Zadanie3** (строки 29–50 на рис. 3.6) содержит описание процедуры численного дифференцирования с регуляризацией на квадратной области определения.

Строки 30–32. Заданы параметры для построения квадратной области определения переменных **s** и **t** для осуществления операции численного дифференцирования, а также вызывается процедура **taskini** для инициализации новой задачи на квадратной сетке по переменным **s** и **t**.

Строка 33. Назначение функции **uval** для использования в расчете. Здесь выбрано **uval** = **Udelta** - **Udelta(1)**, т. е. правая часть $u(t) - u_0$ интегрального уравнения для задачи дифференцирования.

Строки 35–37. Настройка параметров решения СЛАУ для поиска минимума функционала Тихонова: **ZSOLVE** – переключение между процедурами **linsolve(0)** и **backslash(1)**; **SINGULAR** – включение (1) корректировки сингулярности и настройка числового параметра **ACOR** для этой корректировки.

Строка 38. Выполнение расчета матрицы треугольного ядра для преобразования уравнения Вольтерра в интегральное уравнения Фредгольма 1-го рода.

<pre>>> Zadanie3 302 > C_const= 10 > 1; alpha_true= 0; N= 0 Iterat; rho= -9130.2;</pre>	<pre>>> Zadanie3 302 > C_const= 10 > 1; alpha_true= 0.00033729; N= 13 Iterat; rho= 1.9482e-018;</pre>
<pre>s Za 0.025000 -19405.977105 0.075000 95317.630009 0.125000 -150302.849837 0.175000 56568.844763 0.225000 58681.155926 0.275000 -50417.899256 0.325000 46248.961927 0.375000 -84217.970667 0.425000 62691.725657 0.475000 -15218.784612 0.525000 98.725029 0.575000 29.193650 0.625000 2.019380 0.675000 64.154376 0.725000 -193.244987 0.775000 -53.542157 0.825000 163.513517 0.875000 -52.860672 0.925000 12.311383 0.975000 -4.533448 >></pre>	<pre>s Za 0.025000 0.185721 0.075000 0.192208 0.125000 0.205078 0.175000 0.224183 0.225000 0.249331 0.275000 0.280271 0.325000 0.316683 0.375000 0.358168 0.425000 0.404231 0.475000 0.454266 0.525000 0.507544 0.575000 0.563186 0.625000 0.620146 0.675000 0.677183 0.725000 0.732838 0.775000 0.785400 0.825000 0.832872 0.875000 0.872935 0.925000 0.902900 0.975000 0.919668 >></pre>

Рис. 3.8. Пример вывода данных на экран панели Матлаб. Показаны результаты двух расчетов для одного и того же набора входных данных: слева — расчет без регуляризации ($\alpha=0$), справа — расчет с регуляризацией по методу А.Н.Тихонова. Значение параметра регуляризации ($\alpha=3.3729 \times 10^{-4}$) было получено при помощи процедуры **AlphaFinder**

Строки 39–41. Здесь заданы параметры для конечно-разностной аппроксимации решаемой задачи и настройки итерационного процесса: параметр **alpha0**, соответствующий стартовому значению для итерационного расчета параметра регуляризации α ; параметр **Err_alph**, соответствующий значению ε , используемому в модификации метода хорд для указания точности расчета α ; параметр **Auto**, позволяющий включать (**Auto**=1) и выключать (**Auto**=0) автоматический выбор параметра регуляризации α . В последнем случае, в качестве параметра регуляризации используется значение **alpha0**, заданное пользователем.

Строка 43. Вызов внешней процедуры **AlphaFinder**, которая осуществляет расчет в рамках метода регуляризации А.Н.Тихонова.

Строки 45–47. Настройка параметров для графического вывода результатов решения задачи численного дифференцирования на экран панели Матлаб. Пример графического вывода показан на рис. 3.9.

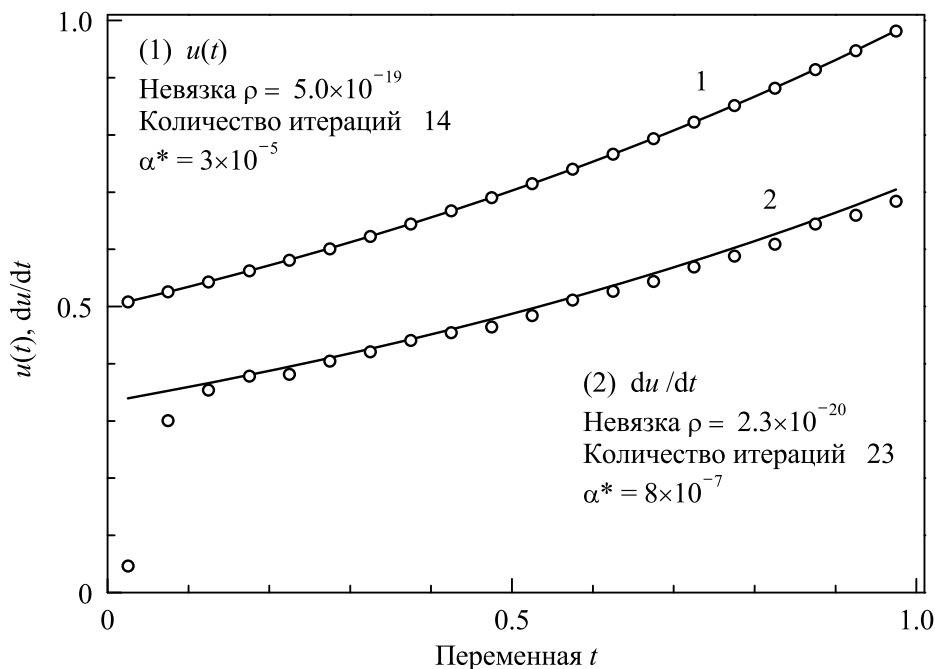


Рис. 3.9. Результаты расчета задачи численного дифференцирования: 1 – функции правой части $u(t)$; 2 – производная du/dt . Сплошные кривые – аналитический расчет U_t и dU_t/dt , светлые кружки – $U_{\Delta t}$ и приближенное решение $z_{\eta}^{\alpha*}$ (результат численного дифференцирования), рассчитанное методом регуляризации А. Н. Тихонова

Строка 48. Вызов процедуры для аналитического расчета функции U_t – значения интеграла в исходном интегральном уравнении. Аналитическая формула размещена в отдельном файле `integral.m`.

Строка 49. Вызов процедуры для аналитического расчета функции $Dudt$ – значения производной $du(t)/dt$ от правой части интегрального уравнения. Аналитическая формула размещена в отдельном файле `derivat.m`.

Строка 50. Вызов процедуры для графического вывода результатов численного дифференцирования на экран панели Матлаб.

3.5. Выполнение численного расчета в среде Матлаб

Цель проведения численного расчета: используя регуляризирующий алгоритм А. Н. Тихонова, найти приближенное решение, удовлетворяющее интегральному уравнению Фредгольма 1-го рода с гладким ядром.

1. Получив индивидуальное задание и используя заданную функцию ядра $K(t, s)$ и заданное точное решение \bar{z} , необходимо рассчитать точное значение правой части уравнения, т. е. функцию

$$\bar{u}(t) = \int_a^b K(t, s) \bar{z}(s) ds.$$

2. Составить определяемые пользователем функции ядра (`yadro`), точного решения (`zbar`), а также аналитические выражения для интеграла (`integral`) и производной (`derivat`). Составленные пользователем функции `yadro`, `zbar`, `integral` и `derivat` следует оформить в виде одноименных файлов с расширением `*.m`, которые следует разместить в рабочей директории среды Матлаб. Если в этой директории уже присутствуют старые файлы `yadro.m`, `zbar.m`, `integral.m` или `derivat.m`, то их следует заменить на новые файлы, составленные пользователем в соответствии с заданием.

3. Внести в шаблон функции `Zadanie3` данные своего индивидуального задания:

- диапазон $[a, b]$ изменения переменной s (строка 04);
- диапазон $[c, d]$ изменения переменной t (строка 05);

4. Задать и внести в шаблон функции `Zadanie3` варьируемые параметры расчетного задания:

- точность задания правой части δ и оператора h (строки 6, 7);
- точность вычисления параметра регуляризации ε (строка 11);
- количество узлов сетки $Nval$ и $Mval$ (строка 14);
- предельное количество итераций $Imax$ (строка 12);

5. Осуществить выбор параметра регуляризации $\alpha > 0$ по схеме обобщенной невязки. Начальное приближение α_0 определить из оценки верхней границы параметра регуляризации $\bar{\alpha}$

$$\bar{\alpha} = \|\hat{\mathbf{A}}_h\| \times \left[h + \frac{\sqrt{h^2 + (\bar{\alpha} + h^2)(C^2 - 1)}}{C^2 - 1} \right], \quad \bar{\alpha} = \frac{\|\hat{\mathbf{A}}_h\|^2 C \delta}{\|u_\delta\| - C \delta},$$

где C – константа, выбранная из условия $\|u_\delta\|/\delta > C > 1$. Рассчитать верхнюю границу $\bar{\alpha}$ параметра регуляризации α и задать значение стартового параметра $\alpha_0 = \bar{\alpha}$.

Выражение для нормы оператора в $L_2[c, d]$ имеет следующий вид:

$$\|\hat{\mathbf{A}}\|_{L_2[c,d]} = \left(\frac{1}{b-a} \int_c^d \left[\int_a^b K(t,s) ds \right]^2 dt \right)^{1/2}.$$

6. Используя модификацию метода хорд, *итеративно* уточнить корень уравнения для обобщенной невязки $\rho(\alpha^*) = 0$.

7. Решая уравнение Эйлера для вычисленного α^* , определить приближенное решение $z_{\eta}^{\alpha^*}$.

8. Используя значения правой части $u(t)$, полученные в п. 1, выполнить расчет первой производной этой функции двумя путями. Во-первых, вычислить производную аналитически. Если при этом правая часть задана численно, то необходимо предварительно провести аппроксимацию правой части каким-либо полиномом. Во-вторых, выполнить численное дифференцирование «возмущенной» функции правой части U_{delta} с регуляризацией по А. Н. Тихонову.

9. Зафиксировать численные результаты расчетов для последующего использования в пояснительной записке.

10. Построить три графические диаграммы. Диаграмма «Решение» должна содержать функции $Z_{\text{bar}}(s)$ и $Z_a(s)$ – решения, которые получаются при работе программы *Zadanie3*. Диаграмма «Правая часть уравнения» должна содержать функции: $U_t(t)$ – результат аналитического вычисления интеграла; $U_{\text{bar}}(t)$ и $U_{\text{delta}}(t)$ – результаты работы программы *Zadanie3*. Диаграмма «Численное дифференцирование» должна содержать функцию правой части U_{delta} и результаты численного и аналитического дифференцирования этой функции

11. Написать обоснованные выводы, где отразить, в частности, следующие результаты выполнения задания:

- рассчитанные параметры к выбору констант C , $\bar{\alpha}$, $\tilde{\alpha}$;
- оптимальное значение параметра регуляризации α и результаты сравнения расчетов без регуляризации ($\text{Auto}=0$, $\alpha_0=0$) и с оптимальным значением параметра регуляризацией ($\text{Auto}=1$);
- минимальный достигнутый уровень функционала невязки;
- сравнение графиков точного (Z_{bar}) и приближенного (Z_a) решений;
- сравнение трех графиков, соответствующих правой части $u(t)$;

3. Регуляризация по методу обобщенной невязки

— сравнение графиков численного дифференцирования правой части.

— влияние вносимого уровня возмущения правой части уравнения на найденное приближенное решение.

12. Оформить пояснительную записку в соответствии с установленными требованиями.

4. Итеративная регуляризация вариационных неравенств

Вариационные методы и вариационные принципы являются важным математическим инструментом, который широко применяется не только в математической физике и прикладной математике, но и для решения сугубо прикладных задач в области физики, техники и технологий. Интерес к вариационным формулировкам задач объясняется рядом причин: многие фундаментальные законы естествознания имеют характер вариационных принципов и вариационные методы часто оказываются весьма эффективным инструментом численного решения многих задач.

Вариационные неравенства являются удобной формой для записи и исследования разнообразных линейных и нелинейных задач. Так, например, в виде вариационного неравенства можно сформулировать задачи, требующие решения нелинейного уравнения, нахождения экстремальной точки какого-либо функционала, нахождения точки равновесия в игре n лиц и многие другие.

4.1. Сведения из теории вариационного исчисления

Зададим некоторый произвольный функционал $J[y]$ и рассмотрим его приращение

$$\Delta J = J[y + h] - J[y],$$

отвечающее приращению h «независимой» переменной y . Если зафиксировать аргумент y , то ΔJ будет представлять собой функционал от независимой переменной h .

Вариацией (дифференциалом) δJ функционала J называется главная линейная часть приращения ΔJ , т. е. линейный функционал $\varphi(h)$, который отличается от ΔJ на бесконечно малую величину порядка выше первого по отношению к норме приращения $\|h\|$. Таким образом,

$$\Delta J = \varphi(h) + \alpha \|h\|,$$

где $\alpha \rightarrow 0$, когда норма приращения $\|h\| \rightarrow 0$.

Функционал $J[y]$ будет достигать экстремума при $y = y_0$ при условии, что $J[y] - J[y_0]$ сохраняет знак в некоторой окрестности кривой

$y_0(x)$.

В вариационном исчислении применяют следующую терминологию в отношении экстремумов (рассмотрим случай задачи на минимум).

Сильный минимум. Принято говорить, что на функции $y_0(x)$ достигается сильный минимум, если $J[y] \geq J[y_0]$ для всякого $y \in E$ из окрестности y_0 (шар с центром в y_0 радиуса r) в метрике $C[a, b]$:

$$\max_{x \in [a, b]} |y(x) - y_0(x)| \leq r, \quad r > 0.$$

Слабый минимум. Для слабого минимума определение формулируется аналогично, но окрестность выбирается в метрике $C^{(1)}[a, b]$:

$$\max_{x \in [a, b]} |y(x) - y_0(x)| + \max_{x \in [a, b]} |y'(x) - y'_0(x)| \leq r, \quad r > 0.$$

В первом случае требуется, чтобы функции $y(x)$ были равномерно «близки» к функции $y_0(x)$, а во втором случае дополнительно требуется «близость» их первых производных. Совершенно очевидно, что если какой-то функционал имеет сильный минимум в точке $y_0(x)$, то он должен иметь и слабый минимум в той же точке. Обратное утверждение в общем случае не является справедливым.

Для дальнейшего обсуждения следует конкретизировать общий вид функционала $J[y]$ и решаемую задачу. Пусть $F(x, y, y')$ – функция, имеющая непрерывные частные производные по всем переменным до второго порядка включительно. Среди всех функций $y(x)$, имеющих непрерывную производную и удовлетворяющую условиям

$$y(a) = A, \quad y(b) = B, \quad (4.1)$$

найти функцию $y = y_0$, при которой имеет место экстремум функционала

$$J[y] = \int_a^b F(x, y, y') dx. \quad (4.2)$$

Функционал такого вида обладает свойством *локальности* (интеграл суммы равен сумме интегралов) и характерен для многих практических задач.

Необходимое условие экстремума. Для того чтобы функционал $J[y]$ при $y = y_0$ достигал экстремума, необходимо, чтобы его дифференциал (если он существует) обращался в нуль при $y = y_0$, т. е.

$$\delta J \equiv 0, \quad \text{при } y = y_0. \quad (4.3)$$

При выполнении этого условия функция y_0 называется экстремалью функционала $J[y]$. Второе необходимое условие экстремума заключается в том, что второй дифференциал должен быть неотрицателен.

Достаточное условие минимума. Чтобы проверить, действительно ли достигается экстремум на найденной экстремали, нужно воспользоваться достаточным условием экстремума, которое состоит из трех (слабый экстремум) или четырех (сильный экстремум) пунктов, выполняющихся одновременно:

- 1) функция $y_0(x)$ является экстремалью функционала $J[y]$;
- 2) для этой функции выполняется усиленное условие Лежандра, которое имеет вид строгого неравенства. В случае задачи на минимум это условие таково

$$F_{y'y'}(x, y_0(x), y_0'(x)) > 0, \quad x \in [x_0, x_1].$$

В случае задачи на максимум знак этого неравенства следует сменить на противоположный: ' $<$ '. Здесь обозначение $F_{y'y'}$ имеет смысл второй частной производной по переменной y' ;

- 3) выполняется усиленное условие Якоби, т. е. отрезок $]a, b]$ не содержит точек, сопряженных точке a . Напомним, что точка $\bar{x} \in]a, b]$ называется сопряженной с точкой $x = a$, если для уравнения Якоби (4.4) существует нетривиальное решение $h(x)$ такое, что $h(a) = 0$ и $h(\bar{x}) = 0$

$$Qh - \frac{d}{dx}(Ph') = 0. \quad (4.4)$$

Здесь использованы общепринятые обозначения Q и P :

$$Q = \frac{1}{2} \left(F_{yy} - \frac{dF_{yy'}}{dx} \right), \quad P = \frac{1}{2} F_{y'y'}.$$

Для определенности на решение $h(x)$ накладывается требование, чтобы функция $h(x)$ удовлетворяла условиям $h(x_0) = 0$ и $h'(x_0) = 1$;

- 4) в случае сильного минимума вдоль экстремали $y_0(x)$ выполнено неравенство $E(x, y, p, y') > 0$ при любых конечных $p \neq y'$, где использовано обозначение $E(x, y, p, y')$ – функция Вейерштрасса.

Простейшая задача вариационного исчисления и уравнение Эйлера. Рассмотрим задачу с закрепленными концами. Пусть $F(x, y, z)$ – функция, имеющая непрерывные частные производные по всем переменным до второго порядка включительно. Среди всех функций $y(x)$, имеющих непрерывную производную и удовлетворяющую условиям

$$y(a) = A, \quad y(b) = B, \quad (4.5)$$

найти ту функцию, при которой имеется экстремум функционала

$$J[y] = \int_a^b F(x, y, y') dx. \quad (4.6)$$

Простейшая задача вариационного исчисления состоит в отыскании *слабого* экстремума функционала вида $J[y]$ на множестве всех гладких кривых, соединяющих две заданные точки.

Из необходимого условия экстремума (4.3) следует утверждение: для того чтобы функционал $J[y]$ (4.6), определенный на множестве функций $y = y(x)$, имеющих непрерывную производную и удовлетворяющих условиям (4.5), достигал на данной функции $y(x)$ экстремума, необходимо, чтобы эта функция удовлетворяла уравнению *Эйлера*

$$F_y - \frac{d}{dx} F_{y'} = 0. \quad (4.7)$$

Здесь F_y и $F_{y'}$ имеют смысл частных производных по переменным y и y' соответственно.

Уравнение Эйлера представляет собой дифференциальное уравнение второго порядка. Его решение должно зависеть от двух произвольных постоянных, которые определяются из двух краевых условий (4.5).

Решения уравнения Эйлера, т. е. интегральные кривые этого уравнения, называются *экстремалими*.

Так как уравнение Эйлера дополняется не начальными, а граничными условиями, то теорема Коши о существовании и единственности решения дифференциального уравнения здесь неприменима. Иными словами, экстремаль не обязательно существует, а если существует, то не обязательно единственна. Все зависит от вида уравнения Эйлера (4.7) и разрешимости системы уравнений для граничных условий (4.5). Отметим, что уравнение Эйлера представляет собой необходимое условие существования экстремума. В этой связи наряду с поиском экстремали (решение уравнения Эйлера) всегда следует проверять также выполнение достаточного условия существования экстремума.

Пример. Вывести уравнение Эйлера для функционала Тихонова. Для упрощения записи опустим обозначения функциональных пространств Z (W_2^1) и U (L_2) при записи норм в функционале Тихонова:

$$M^\alpha[z] = \| \hat{\mathbf{A}} z - u \|^2 + \alpha \| z \|^2,$$

$$\|z\|^2 \equiv \int_a^b [(z(s))^2 + (z'(s))^2] ds = (z, z) + (z', z');$$

$$\|\hat{\mathbf{A}}z - u\|^2 \equiv \int_c^d \left[\int_a^b K(t, s) z(s) ds - u(t) \right]^2 dt = (\hat{\mathbf{A}}z - u, \hat{\mathbf{A}}z - u).$$

Граничные условия: $z'(a) = 0$; $z'(b) = 0$. Посчитаем теперь сильную (Фреше) вариацию функционала Тихонова, т. е. зададим приращение h и выделим линейную по h часть разности $M^\alpha[z + h] - M^\alpha[z]$. Для получения уравнения Эйлера приравняем к нулю линейную часть приращения.

$$M^\alpha[z + h] - M^\alpha[z] = \|\hat{\mathbf{A}}(z + h) - u\|^2 + \alpha \|z + h\|^2 - M^\alpha[z].$$

Граничные условия для непрерывно дифференцируемой функции h :

$$h'(a) = 0; \quad h'(b) = 0.$$

Рассмотрим вспомогательные математические выкладки:

$$\begin{aligned} \|\hat{\mathbf{A}}(z+h) - u\|^2 &= \|(\hat{\mathbf{A}}z - u) + \hat{\mathbf{A}}h\|^2 = \\ &= ((\hat{\mathbf{A}}z - u) + \hat{\mathbf{A}}h, (\hat{\mathbf{A}}z - u) + \hat{\mathbf{A}}h) = \\ &= \|\hat{\mathbf{A}}z - u\|^2 + 2((\hat{\mathbf{A}}z - u), \hat{\mathbf{A}}h) + (\hat{\mathbf{A}}h, \hat{\mathbf{A}}h) = \\ &= \|\hat{\mathbf{A}}z - u\|^2 + 2[\hat{\mathbf{A}}^* \hat{\mathbf{A}}z - \hat{\mathbf{A}}^* u, h] + \|\hat{\mathbf{A}}h\|^2. \end{aligned}$$

Последний член в этом выражении удовлетворяет условию $\|\hat{\mathbf{A}}h\|^2 \leq \|\hat{\mathbf{A}}\|^2 \|h\|^2$, откуда следует, что $\|\hat{\mathbf{A}}h\|^2 = o(\|h\|)$. Далее

$$\begin{aligned} \|z+h\|^2 &= (z+h, z+h) + (z'+h', z'+h') = \\ &= (z, z) + 2(z, h) + (h, h) + (z', z') + 2(z', h') + (h', h') = \\ &= \|z\|^2 + 2[(z, h) + (z', h')] + \|h\|^2. \end{aligned}$$

$$(z', h') = \int_a^b z'(s) h'(s) ds = z' h \Big|_a^b - \int_a^b z''(s) h(s) ds = -(z'', h),$$

где $z' h \Big|_a^b = 0$, т. к. $z'(a) = z'(b) = 0$.

Итак, с учетом вспомогательной выкладки можно записать

$$\begin{aligned} M^\alpha[z+h] - M^\alpha[z] &= 2 \left(\hat{\mathbf{A}}^* \hat{\mathbf{A}} z - \hat{\mathbf{A}}^* u, h \right) + \| \hat{\mathbf{A}} h \|^2 + \\ &\quad + 2\alpha [(z, h) - (z'', h)] + \alpha \| h \|^2 = \\ &= 2(\hat{\mathbf{A}}^* \hat{\mathbf{A}} z - \hat{\mathbf{A}}^* u + \alpha(z - z''), h) + \| \hat{\mathbf{A}} h \|^2 + \alpha \| h \|^2. \end{aligned}$$

Выделим линейную по h часть приращения и приравняем ее нулю:

$$\left(\hat{\mathbf{A}}^* \hat{\mathbf{A}} z - \hat{\mathbf{A}}^* u + \alpha(z - z''), h \right) = \int_a^b \underbrace{\left(\hat{\mathbf{A}}^* \hat{\mathbf{A}} z - \hat{\mathbf{A}}^* u + \alpha(z - z'') \right)}_{=0} h(s) ds = 0.$$

Окончательно получаем уравнение Эйлера в явном виде

$$\hat{\mathbf{A}}^* \hat{\mathbf{A}} z + \alpha(z - z'') = \hat{\mathbf{A}}^* u.$$

4.2. Элементы теории вариационных неравенств

1. Постановка задачи. Зададим некоторое точечно-множественное преобразование $F(x)$, отображающее элементы вещественного банахова пространства B в другое банахово пространство B^* , которое сопряжено с пространством B и задано на подмножестве $D_F \subseteq B$.

Рассмотрим некоторое замкнутое выпуклое множество Q в пространстве B ($Q \subseteq B$). Задача решения вариационного неравенства заключается в нахождении элемента $x^* \in D_F \cap Q$ такого, что

$$(F(x^*), x^* - z) \leq 0, \quad \forall z \in Q. \quad (4.8)$$

Скобки в левой части выражения (4.8) в случае пары B, B^* будут обозначать соответствующий функционал на элементе. В случае, когда пространство B – гильбертово, эти два пространства B и B^* можно отождествить. После такого отождествления выражение в левой части неравенства (4.8) будет соответствовать некоторому скалярному произведению. Целесообразность такого отождествления зависит от конкретной решаемой задачи. Во всех случаях, когда пространство B гильбертово, будем считать, что упомянутое отождествление уже сделано и скобки в выражении (4.8) означают обычное скалярное произведение.

При дальнейшем обсуждении все пространства B будем считать вещественными. Рассмотрение комплексных пространств не представляет принципиальных трудностей, при этом однако становятся более громоздкими все перечисления условий и полученных результатов.

Для точечно-множественного отображения $F(x)$ обозначение $F(x^*)$ в выражении (4.8) будет означать некоторый подходящий элемент образа x^* при отображении F .

В этом случае более точная запись (4.8) выглядит таким образом:

$$\begin{aligned} \exists x^* \in D_F \cap Q \quad \exists^* \in F(x^*): \\ (y^*, x^* - z) \leq 0 \quad \forall z \in Q. \end{aligned}$$

Далее всюду будет использоваться компактная запись выражения (4.8), которую в случае точечно-множественного отображения следует понимать так, как это только что было объяснено выше.

Вариационное неравенство (4.8) считается универсальной формой записи различных задач линейного и нелинейного анализа.

Пусть F – оператор, отображающий элементы пространства B в сопряженное пространство B^* , тогда задача решения нелинейного уравнения

$$F(x) = 0 \tag{4.9}$$

будет эквивалентна задаче решения вариационного неравенства (4.8), причем $Q = B$. Для точечно-множественного отображения F выражение (4.9) следует понимать как включение $0 \in F(x)$.

Задача нахождения минимума функционала $\Phi(x)$ на замкнутом выпуклом множестве $Q \subseteq B$ также эквивалентна задаче решения вариационного неравенства (4.8), если, например, $\Phi(x)$ имеет субдифференциал $\partial\Phi = F$ в каждой точке Q . В этом случае в выражении (4.8) в качестве F следует выбирать субдифференциал $\partial\Phi$ или обычный градиент в дифференцируемом случае.

В частности, если был задан квадратичный функционал невязки

$$\Phi(x) = \| \hat{A}x - u \|_H^2,$$

где \hat{A} – линейный непрерывный оператор, действующий в H , то этот функционал Φ будет непрерывным, выпуклым и имеющим субдифференциал. Задача решения соответствующего вариационного неравенства значительно перекрывает, в частности, задачу линейного уравнения (3.10) при априорных условиях на решение $x \in Q$. Поэтому задачу (3.10), рассмотренную в предыдущем разделе, можно интерпретировать как задачу решения соответствующего вариационного неравенства.

Отметим, что реализуемость и эффективность практически любых итеративных алгоритмов решения неравенства (4.8) существенно зависит от конкретного вида множества Q , так как отдельные шаги численных алгоритмов тесно связаны с проектированием точки на множестве Q , а также с решением на множестве Q некоторых вспомога-

тельных неравенств и т. п. На практике чаще всего применяют способ задания множества Q в виде системы функциональных неравенств:

$$Q = \{x \in B, \quad \varphi_j(x) \leq 0, \quad j = 1, \dots, m\}, \quad (4.10)$$

где $\varphi_j(x)$ – выпуклые непрерывные функционалы в пространстве B .

2. Преобразование к эквивалентным задачам. Во многих случаях исходная формулировка вариационного неравенства (4.8) требует его преобразования к другим эквивалентным задачам, отличающимся от первоначальной по форме и которые могут оказаться более удобными для решения поставленной задачи. Такие преобразования (как и при решении уравнений) оказываются весьма полезными, в частности, при разработке численных схем решения (см. выражение (4.8) в качестве примера).

Одним из путей преобразования решений (4.8), если они существуют, является их аппроксимация последовательностью решений некоторых вспомогательных вариационных неравенств. Одна из схем такой аппроксимации при построении регуляризирующего алгоритма для задач на экстремум функционала была предложена А.Н. Тихоновым. Похожее общее утверждение относительно специальной аппроксимации решений вариационных неравенств было независимо доказано Ф. Браудером. Поэтому аппроксимации такого вида очень часто называют аппроксимациями Браудера–Тихонова. Обсудим вкратце схему аппроксимации вариационного неравенства по методу Браудера–Тихонова.

Определим пару пространств B, B^* таких, что в пространстве B существует всюду определенный полунепрерывный сильно монотонный оператор $\hat{\mathbf{M}}$, т. е. выполняется соотношение

$$(\hat{\mathbf{M}}x - \hat{\mathbf{M}}z, x - z) \leq \|x - z\| \gamma(\|x - z\|) \quad \forall x, z \in B, \quad (4.11)$$

где $\gamma(t)$ – строго возрастающая непрерывная функция на $[0, \infty[$, имеющая предельные точки $\gamma(0) = 0, \lim_{t \rightarrow \infty} \gamma(t) = \infty$.

Для гильбертова пространства B такой оператор заведомо существует и можно, например, положить $\hat{\mathbf{M}} = \hat{\mathbf{E}}, \gamma(t) = t$. В этом случае решения неравенства (4.8) будем приближать решениями неравенств с операторами $F + \varepsilon M, \varepsilon > 0$, и затем устремлять $\varepsilon \rightarrow 0$.

Лемма 6 [10, с. 67–68]. Пусть выполняются условия $D_F \supseteq Q, B$ рефлексивно, F или монотонен и полунепрерывен и $D_F \supseteq B$, или F максимально монотонен и $\text{int } D_F \cap Q \neq \emptyset$.

Тогда существует x_ε – единственное решение вариационного неравенства

$$(F(x) + \varepsilon M(x), x - z) \leq 0, \quad z \in Q, \quad (4.12)$$

и если K – множество решений (4.8) – не пусто, то справедливо предельное соотношение

$$\lim_{\varepsilon \rightarrow 0} \|x_\varepsilon - y\| = 0,$$

где $y \in K$ – единственное решение вариационного неравенства

$$(M(x), x - z) \leq 0 \quad \forall z \in K. \quad (4.13)$$

З а м е ч а н и е 1. Сформулированные выше условия на оператор M допускают значительное ослабление. В частности, вместо однозначности и полунепрерывности M всюду вполне достаточно потребовать, например, максимальной монотонности оператора M и условия $\text{int } D_M \supseteq Q$.

З а м е ч а н и е 2. Если B – рефлексивное пространство, то множество $Q = Q_1 \cap Q_2$, Q_2 можно задать в виде выражения (4.10). Пусть на пространстве B определен функционал вида

$$\psi(x) = \sum_{j=1}^m [\varphi_j(x)]_+^p, \quad p \geq 1. \quad (4.14)$$

В математическом программировании функционал (4.14) является одним из возможных «штрафных» функционалов. Задав на B выпуклый и непрерывный функционал φ_j , получим, что $\psi(x)$ также является выпуклым и непрерывным на B . Поэтому существует субдифференциал $\partial\psi$, являющийся максимальным монотонным оператором. Можно видеть, что множество $Q = Q_1 \cap Q_2$ будет совпадать с множеством решений вариационного неравенства (на Q_1)

$$(\partial\psi(x), x - z) \leq 0, \quad z \in Q_1. \quad (4.15)$$

Запишем аппроксимирующее неравенство (4.12) для выражения (4.15). При этом в качестве M возьмем оператор F :

$$(\varepsilon F(x) + \partial\psi(x), x - z) \leq 0 \quad \forall z \in Q_1. \quad (4.16)$$

Если в основном неравенстве (4.8) оператор F обладает свойством сильной монотонности (4.11), то при сравнительно необременительных условиях непрерывности или максимальной монотонности решение неравенства (4.16) x_ε при $\varepsilon \rightarrow 0$ должно будет аппроксимировать решение (4.8) (которое в данном случае будет единственным!). Основная идея перехода от исходного неравенства (4.8) к аппроксимирующему неравенству (4.16) заключается в том, что в неравенстве (4.16) множество Q_1 устроено намного проще, чем первоначальное множество Q , т.к. часть ограничений была снята. В предельном случае множество Q может просто совпадать со всем пространством, тогда выражение (4.16) превращается в уравнение (включение) $0 \in \varepsilon F(x) + \partial \psi(x)$. Если оператор F является потенциальным, то описанная процедура совпадает по сути с известным в математическом программировании методом штрафа.

4.2.1. Принцип итеративной регуляризации

Рассмотренная выше аппроксимация применяется для «улучшения» первоначальной задачи, чтобы на следующем этапе решения неравенства (4.12) можно было бы, например, применить стандартную итерационную процедуру. Действительно, аппроксимирующий оператор F_ε в неравенстве (4.12), в отличие от оператора F в исходном выражении (4.8), является сильно монотонным и обычно этого бывает достаточно для обеспечения сходимости итераций. Однако при фиксированном $\varepsilon \neq 0$ простое увеличение количества выполненных итераций в итерационной процедуре решения аппроксимирующего неравенства (4.12) в общем случае не приводит к аппроксимации решения исходного выражения (4.8).

При построении итерационной последовательности, которая при увеличении номера итерации k сходилась бы к решению исходного неравенства (4.8), необходимо параллельно построить последовательность параметров ε_k такую, что $\varepsilon_k \rightarrow 0$ при $k \rightarrow \infty$. Процедура построения такой последовательности носит название «итеративной регуляризации». Согласно принципу итеративной регуляризации для начала необходимо выбрать за основу подходящий итеративный метод решения регулярного неравенства (4.12) и далее априорно так назначить последовательность параметров $\varepsilon_k \rightarrow 0$ при $k \rightarrow \infty$, чтобы процедура итеративной регуляризации сходилась (сильно) к решению неравенства (4.8), если это решение существует. Для этого каким-либо образом выбирают стартовые значения параметра ε_0 и точки $z_0 \in Q$. Далее записывают неравенство (4.12) с $\varepsilon = \varepsilon_0$ и делают первый шаг выбранного итеративного метода, предназначенного для решения «ре-

гулярного» неравенства (4.12). Результатом выполнения первого итерационного шага будет точка z_1 . Теперь выбирают значение $\varepsilon = \varepsilon_1$ и с парой ε_1, z_1 осуществляют вторую итерацию, результатом которой будет следующая точка z_2 . Повторяя описанную процедуру получают последовательность точек z_k . После построения численной схемы процедуры итеративной регуляризации необходимо изучить ее сходимость, придерживаясь следующей общей схемы.

Пусть уже выбран итеративный метод, при помощи которого в процедуре итеративной регуляризации будет осуществляться переход от точки z_n к следующей точке z_{n+1} ($n = 0, 1, \dots$). Для краткости будем его называть *базовым итеративным методом*. В общем бесконечном случае для изучения сходимости базового итеративного метода к решению неравенства (4.12) с фиксированным $\varepsilon = \varepsilon_n$ обычно применяют метод функционалов Ляпунова. Известно, что функционал Ляпунова – это такой функционал на изучаемой итеративной последовательности, по поведению которого при возрастании номера итерации k можно судить о сходимости в том или ином смысле рассматриваемых итераций. В обычных случаях в качестве функционала (функции) Ляпунова принято использовать норму разности между точным решением и текущей точкой или какие-либо другие похожие функционалы. Не останавливаясь далее на обсуждении ее явного вида, обозначим функцию Ляпунова через $U(\varepsilon_n, x)$, предполагая для определенности, что имеет место соотношение

$$U(\varepsilon_n, x_{\varepsilon_n}) = 0.$$

Можно далее также записать следующее очевидное неравенство:

$$U(\varepsilon_{n+1}, z_{n+1}) \leq U(\varepsilon_n, z_{n+1}) + |U(\varepsilon_n, z_{n+1}) - U(\varepsilon_{n+1}, z_{n+1})|. \quad (4.17)$$

Для доказательства сходимости базового итерационного метода необходимо записать связь между элементами $U(\varepsilon_n, z_{n+1})$ и $U(\varepsilon_n, z_n)$ в виде

$$U(\varepsilon_n, z_{n+1}) \leq \chi(U(\varepsilon_n, z_n)). \quad (4.18)$$

Второе слагаемое правой части (4.17) можно оценить через $\varepsilon_n, \varepsilon_{n+1}$ и априорно заданные константы и функции, а затем, подставляя выражение (4.18) в неравенство (4.17), получить разностное уравнение для $U_n = U(\varepsilon_n, z_n)$:

$$U_{n+1} \leq \chi(U_n, \varepsilon_n) + \kappa(\varepsilon_n, \varepsilon_{n+1}). \quad (4.19)$$

Учитывая, что явный вид функции Ляпунова должен подбираться для решения обсуждаемой задачи, потребуем, чтобы функция U_n обладала

свойством сильной положительности, т. е. выполнялось соотношение

$$U_n \geq C(\varepsilon_n) \varphi(\|z_n - x_{\varepsilon_n}\|), \quad (4.20)$$

причем $\varphi(\xi) \rightarrow 0$ тогда и только тогда, когда $\xi \rightarrow 0$. Если определить U как расстояние до точного решения, тогда очевидно, что при значениях параметров $C(\varepsilon_n) \equiv 1$ и $\varphi(t) = t$ неравенство (4.20) будет заведомо выполняться. Выражение (4.19) позволяет также оценить скорость стремления функции U_n к нулю при $n \rightarrow \infty$. Если показать, что $\lim_{n \rightarrow \infty} U_n/C(\varepsilon_n) = 0$ и при этом справедлива лемма 6, то отсюда сразу следует, что $\lim_{n \rightarrow \infty} \|z_n - y\| = 0$, что свидетельствует о сильной сходимости исследуемой процедуры итеративной регуляризации. Подытоживая обсуждение, отметим, что основной проблемой здесь является построение таких последовательностей ε_n , $\varepsilon_n \rightarrow 0$, для которых $U_n = o(C(\varepsilon_n))$. Данная численная схема широко применяется для итеративной регуляризации очень многих конкретных базовых методов и, следовательно, для получения итеративных аппроксимаций решения некорректных задач.

4.3. Пример решения простейшей задачи

1. Решить простейшую задачу вариационного исчисления [11]

$$J[y] = \int_a^b F(x, y, y') dx \rightarrow \text{extr}$$

при $a=1$, $b=e$; граничных условиях $y(a)=1$, $y(b)=2$ и функции F :

$$F(x, y, y') = (y')^2 x - y.$$

Прежде всего вычислим частные производные подинтегральной функции: $F_y = -1$, $F_{y'} = 2xy'$, а потом составим уравнение Эйлера

$$F_y - \frac{d}{dx} F_{y'} = -1 - \frac{d}{dx} (2xy') = 0.$$

Перепишем уравнение Эйлера в общепринятом виде

$$2xy'' + 2y' + 1 = 0.$$

Общее решение дифференциального уравнения имеет следующий вид:

$$y(x) = \frac{C_1}{2} \ln(x) - \frac{1}{2}x + C_2, \quad x \in [a, b].$$

Используя граничные условия, находим единственную экстремаль, подозрительную на искомое решение простейшей задачи

$$y_0(x) = \left(\frac{1+e}{2} \right) \ln(x) - \frac{1}{2}x + \frac{3}{2}, \quad x \in [1, e].$$

2. Результаты расчета в Матлаб. На рис. 4.1 представлены результаты расчета (текст и график) простейшей задачи вариационного исчисления, выполненного с использованием программы, приведенной на рис. 4.2.

```
Подынтегральная функция: F= Dy^2*x - y;
Граничные условия: y(1)= 1; y(2.7183)= 2;
Уравнение Эйлера: -2*Dy - 2*D2y*x - 1 = 0;
Общее решение уравнения Эйлера: y(x) = C2 - x/2 + (C1*log(x))/2;
Уравнение экстремали: y(x) = 1.8591409142295*log(x)- 0.5*x+ 1.5;
```

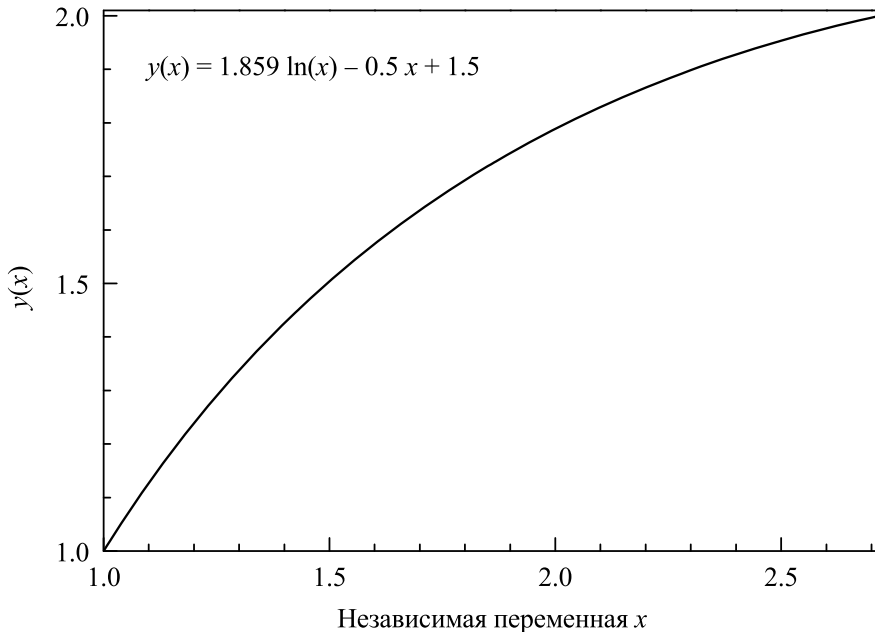


Рис. 4.1. Пример вывода данных (текст и график) на экран панели Матлаб

Текстовая информация:

1. Распечатка подынтегральной функции F , составленной из символьных переменных x , y , Dy .
2. Распечатка граничных условий $y1(x1)$ и $y2(x2)$.
3. Распечатка уравнения Эйлера, составленного из символьных переменных x , y , Dy .

4. Общее решение уравнения Эйлера с двумя неопределенными константами $C1$ и $C2$. Необходимо отметить, что Матлаб поддерживает динамическую нумерацию констант интегрирования, т. е. в каждый конкретный момент времени порядковые номера этих констант могут существенно отличаться от обсуждаемых. Для приведения их к виду $C1$ и $C2$ используется процедура `const_correct`.

5. Уравнение экстремали $y_0(x) = 1.8591409142295 \ln(x) - x/2 + 3/2$. Отметим, что оператор `log`, использованный в Матлаб, имеет смысл оператора натурального логарифма `ln`.

Графическая информация.

На графике (см. рис. 4.1) представлено также уравнение экстремали $y_0(x) = 1.8591 \ln(x) - 0.5x + 1.5$. В табл. 4.1 дано сопоставление коэффициентов этой экстремали с результатами аналитического расчета, проведенного в предыдущем пункте. Из анализа приведенных численных данных (см. табл. 4.1) следует однозначный вывод об их соответствии.

Таблица 4.1

Сравнение коэффициентов экстремали $y_0(x)$

Коэффициент экстремали	Аналитический расчет	Численный расчет в Матлаб
$(C1)/2$	$(1 + e)/2$	1.8591409142295
$(C2)$	$3/2$	1.5

3. Исследование достаточного условия существования слабого экстремума заключается в проверке выполнения трех условий:

1) функция $y_0(x) = 1.8591 \ln(x) - 0.5x + 1.5$ является экстремалью функционала;

2) поскольку $F_{y'y'} = 2x > 0$ при $x \in [1, e]$, то выполняется усиленное условие Лежандра;

3) поскольку $F_y = -1$, то параметры уравнения Якоби $Q = 0$ и $P = x$, а само уравнение Якоби будет иметь вид $-d(xh')/dx = 0$. Единственной зависимостью, удовлетворяющей условиям $h(1) = 0$, $h'(1) = 1$, является кривая $h(x) = \ln(x)$, а значит выполняется усиленное условие Якоби.

На основании теоремы о достаточных условиях экстремума можно утверждать, что $y_0(x)$ доставляет слабый минимум функционалу. Можно дополнительно показать, что экстремаль $y_0(x)$ дает абсолютный минимум в данной задаче.

4. Символьные операции в Матлаб. В состав студенческой версии Матлаб входит *Symbolic Math Toolbox*TM, который содержит функции для обработки и решения символьных математических выражений, а также представления арифметических операций с заданной точностью. В рамках этого пакета можно выполнять аналитическое дифференцирование и интегрирование, решать уравнения, проводить различные упрощения и преобразования выражений, а также конвертировать символьные математические выражения в *m*-коды Матлаб. Для решения простейшей задачи вариационного исчисления используются некоторые команды, реализующие функции символьной обработки и требующие отдельного пояснения. Однако за более детальным описанием этих команд следует обращаться к документации Матлаб.

Символьные вычисления осуществляются над символьными объектами — переменными и константами особого вида. Для их объявления служит команда `syms`, которая в качестве аргументов принимает имена переменных, перечисленные через пробел.

[syms] — сокращение для различных вариантов вызова команды `sym`, осуществляющей конструирование символьных объектов:

```
syms arg1 arg2  является сокращением записи
arg1 = sym('arg1');
arg2 = sym('arg2');

syms arg1 arg2 ... real  является сокращением записи
arg1 = sym('arg1','real');
arg2 = sym('arg2','real');

syms arg1 arg2 ... positive  является сокращением
arg1 = sym('arg1','positive');
arg2 = sym('arg2','positive');

syms arg1 arg2 ... clear  является сокращением записи
arg1 = sym('arg1','clear');
arg2 = sym('arg2','clear');
```

Каждый входной аргумент должен начинаться с буквы и содержать только буквенно-цифровые символы. Символ 'D' и сочетания символов, начинающихся с него, зарезервированы системой символьных вычислений под операторы дифференцирования по независимой переменной.

[diff] — вычисление производной символьного выражения `S` по его свободным переменным, которые определены командой `symvar`.

`diff(S, 't')` или `diff(S, sym('t'))` будет вычислять частную производную выражения `S` по переменной `t`;

`diff(S,n)` будет n раз дифференцировать выражение S по свободным переменным, которые определены командой `syms var`;

`diff(S,'t',n)` будет n раз дифференцировать выражение S по свободной переменной t .

`simplify` — выполняет упрощение символьного выражения S .

`simplify(S)` — пробует несколько алгебраических упрощений для выражения S и возвращает сокращенное представление S . Так,

$$\text{sym}(x), y = \text{simplify}(\sin(x)^2 + \cos(x)^2) = 1.$$

`dsolve` — символьное решение обыкновенных дифференциальных уравнений. Эта команда позволяет находить как общее решение дифференциального уравнения, так и частное его решение, удовлетворяющее заданным начальным или граничным условиям. Заглавная буква D в выражениях трактуется как оператор дифференцирования по независимой переменной, а сочетания Dy , D^2y , D^3y и т. д. обозначают производные соответственно первого, второго и третьего порядков по переменной y .

`dsolve('eqn1', 'eqn2')` — пример вызова команды `dsolve`.

`subs` — операция подстановки (замены) символьных переменных.

`subs(S, 'NEW')` подставляет выражение NEW вместо свободной символьной переменной в выражении S . Рассмотрим пример: из выражения $Q(t) = (t + 2)/(t^2 + 3t + 4)$ требуется получить $P(t) = Q(t)|_{t=t+2}$.

```
>> syms t;
>> Q = (t + 2)/(t^2 + 3*t + 4);
>> P = subs(Q, t, t+2)
P =
(t + 4)/(3*t + (t + 2)^2 + 10)
>> P = collect(P)
P =
(t + 4)/(t^2 + 7*t + 14)
```

Команда `collect` выполняет символьное вычисление коэффициентов при степенях свободной переменной t , используется для упрощения.

`solve` — символьное решение алгебраических и трансцендентных уравнений. Рассмотрим примеры вызова данной процедуры:

```
solve('eqn1', 'eqn2');
solve('eqn1', 'eqn2', 'var1', 'var2');
```


Здесь `eqn1` и `eqn2` соответствуют символьным выражениям или переменным, в которых они содержатся, тогда как `var1` и `var2` обозначают переменные, относительно которых следует разрешить систему уравнений `eqn1 = 0` и `eqn2 = 0`. Отметим, что правая часть любого уравнения (или всех уравнений) этой системы может быть не равна нулю, тогда в состав выражения `eqn1` или `eqn2` следует включить знак равенства и выражение для правой части. Можно использовать альтернативный вариант: правую часть уравнения перенести влево со знаком минус.

```
>> syms t, s;
>> Sol = solve('3*s - 15','3*t - 15 = 3');
>> [Sol.s Sol.t]
[5, 6]
```

`vpa` — задание точности представления (вывода) результатов арифметических вычислений. Подчеркнем, что речь здесь идет только о точности отображения выводимых данных, а не о точности с которой выполняются арифметические операции.

`vpa(S,D)` использует числовой формат "D значащих цифр" вместо текущего формата, заданного командой `digits`. Параметр D является целым или `sym` представлением целого числа.

5. Программное обеспечение Матлаб. На рис. 4.2 представлен листинг программы в *m*-кодах Матлаб, которая была использована для решения описанной выше простейшей задачи вариационного исчисления. Предваряя обсуждение программы, отметим два важных обстоятельства. Во-первых, строки программного кода на рис. 4.2 пронумерованы только для удобства обсуждения, но сами номера строк не являются частью программного кода. Во-вторых, любой текст в промежутке между знаком % и концом текущей строки трактуется как комментарий и не является частью программного кода.

Строка 0. Это заголовок программы-функции `Zadanie4()`. Пустые скобки означают отсутствие передаваемых извне входных параметров.

Строки 1–2. Очистка памяти рабочей области Матлаб от случайных данных, оставшихся от решения предыдущих задач и очистка экрана панели Матлаб.

Строки 3–4. Задание 14-значного формата отображения действительных чисел и декларирование символьных переменных:

$$x, \quad y, \quad Dy = y' \equiv \frac{dy}{dx}, \quad D2y = y'' \equiv \frac{d^2y}{dx^2}.$$

была использована процедура на основе встроенной команды `diff`:

$$\frac{dF_{y'}}{dx} = \frac{\partial F_{y'}}{\partial x} + \frac{\partial F_{y'}}{\partial y} y' + \frac{\partial F_{y'}}{\partial y'} y''.$$

Строки 13–14. Составление левой части дифференциального уравнения Эйлера и применение процедуры `simplify` для ее упрощения. Дифференциальное уравнение необходимо представить в виде строковой переменной для последующего его решения. Преобразуем символьную переменную `Euler` в строку и добавим в конце равенство нулю.

Строки 15–19. Команда `dsolve` позволяет находить как общее решение дифференциального уравнения, так и частное его решение, удовлетворяющее заданным начальным или граничным условиям. Дополнительная процедура `const_correct` была использована для переименования констант интегрирования в `C1` и `C2`. Найдем вначале общее решение уравнения, а потом подставим в него граничные условия. Решаем уравнение Эйлера и проверяем, является ли решение единственным. Программа будет работать дальше, если решение существует и единственное. Если нет, прекращаем работу программы.

Строки 21–24. Подставим в найденное аналитическое решение `Sol1` граничные точки `x1` и `x2`, преобразованные в символьные переменные. Затем преобразуем эти символьные выражения в строки и приравняем их соответственно `y1` и `y2`.

Строка 25–28. Решаем полученную систему уравнений – при помощи процедуры `solve` находим значения произвольных постоянных `C1` и `C2`.

Строка 28–29. Подставляем найденные константы в аналитическое решение `Sol1` и упрощаем получившееся выражение для решения системы уравнений. Ограничим полученное решение 14 знаками – этого вполне достаточно для построения графика решения.

Строка 31–32. Задаем нижний (`Dlim1`) и верхний (`Dlim2`) пределы по оси ординат для вывода графика экстремали. В случае монотонно возрастающей функции их можно приравнять значениям `y1` и `y2` краевой задачи. Однако возможна и более точная настройка этих пределов. Например, в случае выпуклой функции необходимо использовать минимальное (`ymin`) и максимальное (`yтах`) значения функции $y(x)$, т. е. `Dlim1 = ymin` и `Dlim2 = yтах`.

Строка 33. Вызов процедуры текстового и графического вывода результатов решения простейшей задачи вариационного исчисления. Данная процедура реализована в виде файла `printout4.p`, находящегося в рабочей директории.

4.4. Выполнение практического расчета в среде Матлаб

Цель проведения численного расчета: используя возможности пакета *Symbolic Math Toolbox*TM по обработке символьных выражений в программной среде Матлаб, решить простейшую задачу вариационного исчисления и получить в явном виде уравнение экстремали, т. е. для функционала

$$J[y] = \int_a^b F(x, y, y') dx,$$

где $a, b, y(a), y(b)$ – заданные граничные условия, найти аналитическое выражение для подынтегральной функции $y_0 = F(x, y, y')$ (экстремали) на которой функционал $J[y]$ достигает своей точной нижней грани.

1. Получив индивидуальное задание и используя заданную функцию точного уравнения экстремали $y = F(x, y, y')$, необходимо вручную рассчитать частные производные $F_y, F_{y'}$, составить уравнение Эйлера

$$F_y - \frac{d}{dx} F_{y'} = 0$$

и получить его общее решение с двумя неопределенными константами C1 и C2.

2. Используя граничные условия индивидуального задания, определить значения констант C1, C2 и найти единственную экстремаль, подозрительную на искомое решение простейшей задачи.

3. Используя текстовый файл программы `Zadanie4()` в качестве шаблона, составить вариант программы для решения индивидуального задания в среде Матлаб. Выполнить решение задачи в среде Матлаб, зафиксировать текстовые и графические результаты решения для последующего использования в пояснительной записке.

4. Выполнить исследование достаточного условия существования слабого экстремума, т. е. надо убедительно обосновать выполнение трех пунктов теоремы: функция y_0 является экстремалью функционала $J[y]$; выполняется усиленное условие *Лежандра*; выполняется усиленное условие *Якоби*.

5. Используя найденную формулу экстремали y_0 , определить значение минимума функционала $J(y_0)$.

6. Написать обоснованные выводы, где отразить, в частности, следующие результаты выполнения задания:

- сравнить выражения для экстремалей, полученные аналитическим путем и в результате расчета в среде Матлаб. Если два полученных выражения различаются, то необходимо исследовать их идентичность аналитически, построить и сравнить их графики на одной диаграмме, обсудить причины различия;

- сравнить значения коэффициентов экстремали, полученные аналитическим путем и в результате расчета в среде Матлаб, обсудить результаты сравнения;

- на основании теоремы о достаточных условиях экстремума обсудить выполнение достаточного условия существования экстремума для индивидуальной задачи, обосновать для полученной экстремали классификационные признаки минимума (сильный, слабый, абсолютный);

- рассчитанное значение минимума функционала $J(y_0)$.

7. Оформить пояснительную записку в соответствии с установленными требованиями.

Техническое замечание. Тестирование программного обеспечения было выполнено с использованием версий Матлаб-2010/13. Ввиду того, что разработчик пакета *Symbolic Math Toolbox*TM непрерывно работает над развитием и совершенствованием своего программного продукта, постоянно внося различные изменения в последующие его версии, предложенный файл-шаблон решения задачи (см. рис. 4.2) может быть несовместим с версиями Матлаб, отличными от Матлаб-2010/13. В этом случае читателю предоставляется возможность использовать файл-шаблон решения задачи (см. рис. 4.2) для уяснения алгоритма решения задачи и самостоятельной адаптации программы в *m*-кодах (см. рис. 4.2) к особенностям используемой версии Матлаб. Для получения информации об используемой версии Матлаб необходимо выполнить команду `ver`.

Список библиографических ссылок

1. Огородников И. Н. Введение в обратные задачи физической диагностики : специальные главы высшей математики для технологов. — Екатеринбург : Изд-во Урал. ун-та, 2017. — 199 с.
2. Гольштейн Е. Г. Теория двойственности в математическом программировании и ее приложения. Современные проблемы математики. — М. : Наука, 1971. — 352 с.
3. Линейное и нелинейное программирование / И. Н. Ляшенко, Е. А. Карагодова, Н. В. Черникова, Н. З. Шор. — Киев : Выща школа, 1975. — 372 с.
4. Шапкин А. С., Шапкин В. А. Математические методы и модели исследования операций. — М. : Дашков и К⁰, 2013. — 400 с.
5. Волков И. К., Загоруйко Е. А. Исследование операций : учеб. для вузов. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2000. — 436 с.
6. Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. — М. : Физматлит, 2004. — 572 с.
7. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. — М. : Наука, 1979. — 285 с.
8. Численные методы решения некорректных задач / А. Н. Тихонов, А. В. Гончарский, В. В. Степанов, А. Г. Ягола. — М. : Наука, 1990. — 231 с.
9. Гарантированная точность решения систем линейных уравнений в евклидовых пространствах / С. К. Годунов, А. Г. Антонов, О. П. Кирилук, В. И. Костин. — Новосибирск : Наука, 1998. — 360 с.
10. Бакушинский А. Б., Гончарский А. В. Итеративные методы решения некорректных задач. — М. : Наука, 1989. — 128 с.
11. Краснов М. Л., Макаренко Г. И., Киселев А. И. Вариационное исчисление. — М. : Наука, 1973. — 191 с.

Алфавитный указатель

А

Алгоритм
 преобразования матрицы 23
 приближенного решения 43
 регуляризирующий 30, 31
 симплекс-метода 14
 устойчивый 37

Аппроксимация
 Браудера—Тихонова 98
 конечно-разностная 37
 функционала 75

Б

Базис
 Жорданов 16
 переменных 13
Браудер Ф. 98

В

Вариация
 Фреше (сильная) 95
 полная 41
 функции 41
 функционала 91
Векторы
 выпукло независимые 44
Выпуклая комбинация 43
Выпуклый многогранник 44

Д

Дифференциал функционала 91

З

Задача
 Коши 78
 двойственная 7
 корректная по Тихонову 35
 несовместная 31
 простейшая 102
 прямая 7
 эквивалентная 7
 экстремальная 7
Закрепленный конец 42

И

Информация
 априорная 33, 38
 качественная 37, 46

К

Квазирешение 35
Класс корректности 35
Колмогоров А.Н. 33
Команда Матлаб
 backslash 78
 collect 106
 diff 106
 dsolve 106
 eval 126
 int 126
 linprog 27
 linsolve 78
 linspace 125
 quad 123
 simplify 106
 solve 106
 subs 106
 syms 105
 trapz 125
 ver 111
 vpa 107
Компакт 33, 35, 67

М

Метод
 Гаусса-Жордана 13, 76
 базовый итеративный 101
 интегрирования
 Симпсона 123
 адаптивной квадратуры 123
 прямоугольников 75
 трапеций 125
 квадратного корня 76
 квазирешения 33
 подбора 33
 проекции градиентов 37
 прямоугольника 19
 разложения Холецкого 76

симплекс 8, 12
условного градиента 37, 49, 56
хорд модифицированный 71
штрафа 100

Минимум
сильный 92
слабый 92

Множество
компактное 33
корректности 35, 36

Н

Неравенство
аппроксимирующее 99
вариационное 96–98

О

Оператор
инъективный 34
полунепрерывный 98
регуляризирующий 67
самосопряженный 69
сильно монотонный 98
сопряженный 69

Отображение
точечно-множественное 96

Оценка априорная 37

П

Параметр регуляризации 70
Перебор направленный 14
Погрешность среднеквадратическая 54
Принцип
итеративной регуляризации 100
обобщенной невязки 71

Производная
Фреше (сильная) 49
полная 109
частная 92

Пространство
Соболева 72
афинное 43
банахово 96
гильбертово 69
евклидово 33
компактное 33
метрическое 33
рефлексивное 99
топологическое 33

Р

Регуляризация
Тихонова 67
итеративная 102
на компактах 33

Решение
базисное 13
допустимое 13, 16
опорное 13
особое 13
приближенное 37
регуляризированное 74

С

Свойство
локальности 92
полунепрерывности 98

Сопряженная точка 93

Стабилизатор 68

Столбец
ведущий 16
свободных членов 20

Строка ведущая 16

Субдифференциал 97

Т

Теорема
Коши 94
Тихонова 35
Хелли 37, 40
двойственности 11
фундаментальная
симплекс-метода 12

Тихонов А.Н. 35, 98

У

Уравнение
Вольтерра 78, 86
Фредгольма 47, 63, 72, 78, 79, 86
Эйлера 69, 89, 94
Якоби 93

Условие
Лежандра 93
Липшица 49
Якоби 93
достаточное 93
необходимое 92

Ф

Форма

каноническая 15

Формула

аналитическая 62

аппроксимационная 52

квадратурная 75

прямоугольника 19

Функционал

Ляпунова 101

Тихонова 67, 94

невязки 36, 43

сглаживающий 67

стабилизирующий 68

штрафной 99

Функция

Вейерштрасса 93

Ляпунова 101

встроенная 105

кусочно-гладкая 72

ограниченной вариации 41

Э

Экстремаль 94

функционала 93

Элемент

ведущий 16

Я

Ядро

гладкое 47, 63, 72, 79, 88

единичное 78

треугольное 78, 86

Приложение 1. Упражнения для расчетов в среде Матлаб

Тема «Оптимизация и теория двойственности»

1. Применяя двойственный симплексный метод, решить задачу

$$Z_{\max} = 5x_1 - x_2 - 4x_3$$

при ограничениях

$$\begin{cases} -x_2 + 2x_3 \geq 9, \\ -x_1 + x_2 \geq 1, \\ x_1 + x_2 - 3x_3 \geq 8, \\ x_1 - x_3 \leq 4, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

2. Применяя двойственный симплексный метод, решить задачу

$$Z_{\min} = 2x_1 + 3x_2 + \frac{5}{2}x_3$$

при ограничениях

$$\begin{cases} 2x_1 + x_2 + 3x_3 \geq 6, \\ 2x_1 + 4x_2 + 3x_3 \geq 16, \\ 3x_1 + 4x_2 + 2x_3 \geq 12, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

3. Применяя двойственный симплексный метод, решить задачу

$$Z_{\min} = 4x_1 + 6x_2 + 3x_3$$

при ограничениях

$$\begin{cases} 3x_1 + x_2 + 2x_3 \geq 9, \\ x_1 + 2x_2 + 2x_3 \geq 8, \\ x_1 + 6x_2 \geq 12, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

4. Найти минимальное значение функции

$$Z = -2x_1 + x_2 + 5x_3$$

при ограничениях

$$\begin{cases} x_1 + x_2 - x_3 \leq 4, \\ x_1 - 5x_2 + x_3 \geq 5, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

5. Найти минимальное значение функции

$$Z = 3x_1 + 2x_2 - 4x_3$$

при ограничениях

$$\begin{cases} x_1 + x_2 - 2x_3 \geq 4, \\ 3x_1 - x_2 + 4x_3 = -7, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

6. Найти минимальное значение линейной функции

$$Z = x_1 + 2x_2 + 3x_3$$

при ограничениях

$$\begin{cases} 2x_1 + 2x_2 - x_3 \geq 2, \\ x_1 - x_2 - 4x_3 \leq -3, \\ x_1 + x_2 - 2x_3 \geq 6, \\ 2x_1 + x_2 - 2x_3 \geq 3, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

7. Найти максимальное значение линейной функции

$$Z_{\max} = x_1 + 10x_2 + 8x_3$$

при ограничениях

$$\begin{cases} x_1 + 4x_2 + x_3 = 2, \\ x_1 + 2x_2 - x_3 = 0, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

8. Найти минимальное значение линейной функции

$$Z_{\min} = 4x_1 - x_2 - x_3 + x_4$$

при ограничениях

$$\begin{cases} x_1 + x_2 - 2x_3 + 3x_4 = 1, \\ 2x_1 - x_2 - x_3 + 3x_4 = 2, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3, 4).$$

Записать оптимальные планы исходной и двойственной задач.

9. Найти максимальное значение линейной функции

$$Z_{\max} = x_1 + x_2 + x_3 + x_4$$

при ограничениях

$$\begin{cases} x_1 + 3x_2 + x_3 + x_4 = 5, \\ 2x_1 - x_3 + x_4 = 1, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3, 4).$$

Записать оптимальные планы исходной и двойственной задач.

10. Найти максимальное значение линейной функции

$$Z_{\max} = x_1 + x_2 + x_3 + x_4$$

при ограничениях

$$\begin{cases} x_1 + 3x_2 + 7x_3 - x_4 = 6, \\ x_1 - x_2 - x_3 + 3x_4 = 2, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3, 4).$$

Записать оптимальные планы исходной и двойственной задач.

11. Найти максимальное значение линейной функции

$$Z_{\max} = x_1 + 7x_2 - x_3$$

при ограничениях

$$\begin{cases} x_1 - x_2 - 2x_3 = -1, \\ x_1 + 2x_2 + 13x_3 = 14, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

12. Найти минимальное значение линейной функции

$$Z_{\min} = x_1 - 2x_2 + 3x_3$$

при ограничениях

$$\begin{cases} -2x_1 + x_2 + 3x_3 = 2, \\ 2x_1 + 3x_2 + 4x_3 = 1, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

13. Найти минимальное значение линейной функции

$$Z_{\min} = 3x_1 + x_2 + x_3 + x_4$$

при ограничениях

$$\begin{cases} -2x_1 + 2x_2 + x_3 = 4, \\ 3x_1 + x_2 + x_4 = 6, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3, 4).$$

Записать оптимальные планы исходной и двойственной задач.

14. Найти максимальное значение линейной функции

$$Z_{\max} = 2x_1 + 8x_2 + 2x_3$$

при ограничениях

$$\begin{cases} 4x_1 - x_2 + 3x_3 \leq 7, \\ 7x_1 + 5x_2 + 12x_3 = 19, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

15. Найти минимальное значение линейной функции

$$Z_{\min} = x_1 - 3x_2 + x_3$$

при ограничениях

$$\begin{cases} x_1 + 4x_2 + 3x_3 \leq 12, \\ -x_1 + 2x_2 - 2x_3 \leq 4, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

16. Найти максимальное значение линейной функции

$$Z_{\max} = x_1 + 4x_2 - x_3$$

при ограничениях

$$\begin{cases} x_1 - x_2 + x_3 = 3, \\ 2x_1 - 5x_2 - x_3 = 1, \end{cases} \quad x_j \geq 0 \quad (j = 1, 2, 3).$$

Записать оптимальные планы исходной и двойственной задач.

Тема «Регуляризация на специальных множествах»

№ п/п	Функция ядра $K(t, s)$	Точное решение $\bar{z}(s)$	Границы областей			
			a	b	c	d
1	$\sin^2(ts)$	$1 - s^2$	0	1	0	2
2	$\frac{1}{1 + (t + s)^2}$	$1 - s^2$	0	1	0	2
3	$\exp(ts)$	$1 - s^2$	0	1	-1	1
4	$s \exp(ts)$	$1 - \ln^2(s + 1/2)$	0	1	-1	1
5	$\frac{\exp(ts)}{(t + s)^2}$	$\sin(s)$	0	π	$\pi/2$	π
6	$1 + \sin(ts) + \ln(1 + s)$	$\sin(s)$	0	π	$\pi/2$	π
7	$\sin(ts) + \ln(1 + s)$	$1 - 2 \cos^2(s)$	$\pi/4$	$3\pi/4$	$\pi/2$	π
8	$\sin(ts)(t + s)^2$	$\cos(s)$	0	$\pi/2$	$\pi/2$	π
9	$\cos(ts)(t + s)^2$	$2 \sin^2(s) - 1$	$\pi/4$	$3\pi/4$	$\pi/2$	π
10	$\sin(ts)$	$4s(1 - s)$	0	1	0	2
11	$\cos(ts)$	$4s(1 - s)$	0	1	0	2
12	$\frac{1}{1 + 500(t - s)^2}$	$G(s) - 0.052$	0	1	-2	2
13	$\frac{\exp(ts)}{(t + s)^2}$	$1.35 - \exp(s - 1)$	0	1.3	1	2
14	$\frac{ts}{(t + s)^2}$	$1 - 2 \ln(s + 1/2)$	0	1	1	2
15	$\frac{ts}{\cos(ts)}$	$1 - 2 \cos^2(s)$	$\pi/4$	$3\pi/4$	$\pi/2$	π

Примечание. Функция $G(s) = 0.955 \left[\exp\left(-\frac{(s - 0.3)^2}{0.03}\right) + \exp\left(-\frac{(s - 0.7)^2}{0.03}\right) \right]$.

Тема «Регуляризация по методу обобщенной невязки»

№ п/п	Функция ядра $K(t, s)$	Точное решение $\bar{z}(s)$	Границы областей			
			a	b	c	d
1	$\sin^2(t s)$	s^2	0	1	0	π
2	$[1 + (t + s)^2]^{-1}$	s^2	0	1	0	1
3	$\exp(t s)$	$1 + s^2$	0	1	-1	1
4	$s \exp(t s)$	$1 + \ln^2(s + 1)$	0	1	-1	1
5	$(t + s)^{-2} \exp(s t)$	$\sin(s)$	0	π	$\pi/2$	π
6	$1 + \sin(t s) + \ln(s + 1)$	$\sin(s)$	0	π	$\pi/2$	π
7	$\sin(t s) + \ln(s + 1)$	$3/2 + \cos(s)$	0	π	$-\pi/2$	$\pi/2$
8	$\sin(t s) (s - t)^2$	$\cos^2(s)$	0	π	$-\pi/2$	$\pi/2$
9	$\cos(t s) (s + t)^2$	$\sin^2(s)$	0	π	$-\pi/2$	$\pi/2$
10	$\sin(t s)$	s^3	0	π	$-\pi/2$	$\pi/2$
11	$\cos(t s)$	$1 + s + s^2 + s^3$	0	π	$-\pi/2$	$\pi/2$
12	$s t \cos^{-1}(t s)$	$1 + s + s^2 + s^3$	0	2	-1	1
13	$s t (t + s)^{-2}$	$s + \ln(s + 2)$	0	1	0.5	1
14	$t (t + s)^{-2}$	$s / \ln(s + 1)$	1	2	-1	1
15	$\exp(s t) (t + s)^{-2}$	$\exp(s)$	1	2	-1	1

Тема «Итеративные методы регуляризации»

№ п/п	Подынтегральная функция $F(x, y, y')$	Граничные условия			
		a	b	$y(a)$	$y(b)$
1	$(y')^2 + y^2 + x y$	0	1	0	0
2	$4 y \sin(x) - y^2 - (y')^2$	0	1	0	0
3	$(y')^2 - 4 y^2$	0	$\pi/4$	0	1
4	$(y')^2 - y^2 - x y$	0	$\pi/2$	0	0
5	$(y')^2 - y^2 + 4 y \cos(x)$	0	$\pi/2$	0	$\pi/2$
6	$6 y \sin(2x) + y^2 - (y')^2$	0	$\pi/2$	0	0
7	$(y')^2 + y^2 + 6 y \operatorname{sh}(2x)$	0	1	0	0
8	$\sqrt{1 + (y')^2}/y$	0	1/2	1	$\sqrt{3}/2$
9	$x (y')^2 + 2 y$	1	e	1	2
10	$2 y - x^2 (y')^2$	1	e	e	0
11	$x y' + (y')^2$	0	2	1	0
12	$y' (1 + x^2 y')$	-1	2	1	4
13	$y' (1 + x^2 y')$	1	2	3	5
14	$x^2 (y')^2 + 12 y^2$	1	2	1	8
15	$4 y^2 - (y')^2 + 8 y$	0	$\pi/4$	-1	0

Приложение 2. Вычисление интегралов в среде Матлаб

Вычислительная среда Матлаб имеет мощные средства для выполнения разнообразных вычислительных процедур как численными методами, так и путем символьных вычислений. В данном разделе представлена краткая информация о некоторых возможностях Матлаб по вычислению определенных и неопределенных интегралов, которые рекомендуются к использованию при выполнении расчетных заданий 2—4.

Численное вычисление определенных интегралов

1. Метод адаптивной квадратуры Симпсона. Если подынтегральная функция непрерывна и задана аналитической формулой, то для расчета определенного интеграла можно применить процедуру `quad`, реализующую при интегрировании численный метод адаптивной квадратуры Симпсона.

`Q = quad(fun, a, b)` вычисляет значение интеграла скалярной функции `fun` на интервале $[a, b]$ изменения переменной интегрирования. Процедура использует рекурсивный метод адаптивной процедуры Симпсона и по умолчанию обеспечивает погрешность интегрирования 10^{-6} . Здесь `fun` – указатель на подынтегральную функцию, `a` и `b` – пределы интегрирования. В качестве входного аргумента функция `Y = fun(X)` принимает вектор `X` и возвращает вектор `Y`.

`Q = quad(fun, a, b, tol)` вычисляет значение интеграла скалярной функции `fun` на интервале $[a, b]$ изменения переменной интегрирования. Однако в этом случае вместо значения погрешности интегрирования по умолчанию (10^{-6}) процедура использует значение погрешности, которое задано аргументом `tol`. Очевидно, что при уменьшении значения заданной погрешности интегрирования будет возрастать время расчета интеграла.

Рассмотрим примеры программного кода. Для начала зададим тестовую подынтегральную функцию

$$y(x) = 2x^2 + 3x + 4.$$

Для этого в отдельный файл с именем `myfunc.m` помести программный код, реализующий вычисление выходного вектора Y , содержащего значения функции $y(x)$, для входного векторного аргумента X :

```
function y = myfunc(x)
    y = 2*x.^2 + 3*x +4;
    return;
end
```

При записи аналитической формулы подынтегральной функции необходимо использовать матричные операции с точкой: `.*`, `./` и `.^`.

После этого в окне Матлаб можно задать тестовый программный код и получить результат выполнения процедуры интегрирования:

```
>> a = 0; b = 1;
>> Q = quad(@myfunc, a, b)
Q =
    6.1667
```

В каждой строке программному коду предшествует символ `>>`, который выдается компьютером и означает приглашение (промпт) к вводу программного кода. Строки без промпта соответствуют результатам выполнения предшествующей команды. При обсуждении будут использованы ссылки на номера строк программного кода, при этом нумерации подлежат только строки, содержащие приглашение (промпт).

Зададим для примера подынтегральную функцию с параметром t :

$$y(x) = 2(x - t)^2 + 3(x - t) + 4,$$

программный код которой разместим в файле с именем `myfunc1.m`:

```
function y = myfunc1(x, t)
    y = 2*(x - t).^2 + 3*(x - t) + 4;
    return;
end
```

После этого в окне Матлаб можно задать программный код и получить результат выполнения процедуры интегрирования:

```
>> a = 0; b = 1; t = 2;
>> Q1 = quad(@(x)myfunc1(x, t), a, b)
Q1 =
    4.1667
```

Процедура `quad` не требует предварительной векторизации подынтегральной функции и рекомендуется к использованию в случае аналитически заданной подынтегральной функции.

2. Интегрирование методом трапеций. В случае, когда подынтегральная функция задана численно в виде векторов X (аргумент функции) и Y (значение функции), то для интегрирования можно использовать процедуру `trapz`, которая реализует численное интегрирование методом трапеций. В случае аналитически заданной функции эту процедуру также можно использовать, но необходима предварительная векторизация функции.

$Q2 = \text{trapz}(X, Y)$ вычисляет значение интеграла от функции, численно заданной вектором Y , с аргументом, заданным вектором X . Вектора X и Y должны иметь одинаковую длину, однако шаг векторизации по аргументу ($h_j = x_j - x_{j-1}$) может быть произвольным и переменным. Интегрирование идет всему интервалу изменения переменной x , определенному вектором X .

$Q3 = \text{trapz}(Y)$ соответствует предыдущему случаю, однако здесь используется постоянный шаг векторизации $h = 1$. В случае не единичного постоянного шага векторизации (h) значение интеграла получают путем умножения $h * Q3$.

В окне Матлаб зададим тестовый программный код и получим результат выполнения процедуры интегрирования:

```
>> a = 0; b = 1; N = 100;
>> X = linspace(a, b, N);
>> Y = myfunc(X);
>> h = (X(N) - X(1))/(N - 1);
>> Q2 = trapz(X, Y)
Q2 = 6.1667
>> Q3 = trapz(Y);
>> Q4 = h*Q3
Q4 = 6.1667
```

Здесь первые три строчки программного кода осуществляют процедуру векторизации подынтегральной функции `myfunc`: a и b – пределы интегрирования $x \in [a, b]$; N – количество точек векторизации. В четвертой строке вычисляют постоянный шаг векторизации h .

Символьное вычисление интегралов

1. Вычисление неопределенного интеграла (первообразной).

$Q5 = \text{int}(F, s)$ осуществляет символьное вычисление неопределенного интеграла (первообразной) от подынтегральной функции F , относительно переменной интегрирования s . Здесь s – скалярная пе-

ременная интегрирования (тип `sym`); F – подынтегральная функция (тип `sym`), которая может быть задана с параметрами. В примере – это параметр t .

В командном окне Матлаб зададим тестовый программный код и получим результаты выполнения процедуры интегрирования:

```
>> syms s t
>> F = myfunc(s);
>> Q6 = int(F, s)
Q6 = (s*(4*s^2 + 9*s + 24))/6
>> F1 = myfunc1(s, t);
>> Q7 = int(F1, s)
Q7 = (2*s^3)/3 + (3/2 - 2*t)*s^2 + (2*t^2 - 3*t + 4)*s
>> t = 1;
>> Q8 = eval(Q7)
Q8 = (2*s^3)/3 - s^2/2 + 3*s
```

Строки программного кода (с промптом) соответствуют выполнению следующих команд: строка 1 – определение списка символьных переменных (используется `syms`); строка 2 – задание подынтегральной функции F без параметра; строка 4 – задание подынтегральной функции $F1$ с параметром t ; строка 6 – определение численного значения параметра t ; строка 7 – вычисление выражения $Q7$ при заданном значении параметра $t = 1$ (используется процедура `eval`).

2. Вычисление определенного интеграла.

$Q5 = \text{int}(F, s, a, b)$ осуществляет символьное вычисление определенного интеграла от подынтегральной функции F по ds .

В командном окне Матлаб зададим тестовый программный код и получим результаты выполнения процедуры интегрирования:

```
>> syms s t
>> a = 0; b = 1;
>> F = myfunc(s);
>> Q9 = int(F, s, a, b)
Q9 =
    37/6
>> F1 = myfunc1(s, t);
>> Q10 = int(F1, s, a, b)
Q10 =
    2*t^2 - 5*t + 37/6
```

Учебное издание

Огородников Игорь Николаевич

**Введение в обратные задачи
физической диагностики.
Модельные расчеты в Матлаб**

Редактор О. В. Протасова
Верстка И. Н. Огородникова

Подписано в печать 30.11.2017. Формат 70×1001/16.
Бумага писчая. Цифровая печать. Гарнитура Antiqua.
Усл. печ. л. 11,3. Уч.-изд. л. 10,3. Тираж 100 экз. Заказ 355.

Издательство Уральского университета
Редакционно-издательский отдел ИПЦ УрФУ
620049, Екатеринбург, ул. С. Ковалевской, 5
Тел.: 8 (343) 375-48-25, 375-46-85, 374-19-41
E-mail: rio@urfu.ru

Отпечатано в Издательско-полиграфическом центре УрФУ
620083, Екатеринбург, ул. Тургенева, 4
Тел.: 8 (343) 358-93-06, 350-58-20, 350-90-13
Факс: 8 (343) 358-93-06
<http://print.urfu.ru>



ОГОРОДНИКОВ ИГОРЬ НИКОЛАЕВИЧ

Доктор физико-математических наук, профессор кафедры экспериментальной физики Физико-технологического института УрФУ.

Более 30 лет занимается теоретическими и экспериментальными исследованиями динамики электронных возбуждений, локализованных состояний, дефектов, люминесценции и термостимулированных процессов в широкозонных диэлектриках – оптических материалах для современной коротковолновой лазерной техники, нелинейной и интегральной оптики.

Автор более 450 научных публикаций, 6 изобретений, 23 учебно-методических изданий.

Член Редакционной коллегии международного научного журнала «Radiation Measurements» (Elsevier).